DFACTT DATA FUSION TOOLS & TECHNIQUES

REPORT ON ADVANCED TECHNOLOGIES

VOLUME 1

24 March 1995

Prepared for:

Mr. D. Elsaesser
Defence Research Establishment
Electronic Warfare Division
Ottawa, Ontario

Prepared by:

Software Kinetics Ltd.
65 Iber Road
Stittsville, Ontario Canada
K2S 1E7

Software Kinetics Document No. 1600-116-006 Version 01

105-02577/

DFACTT Data Fusion Tools & Techniques
Report on Advanced Technologies
Volume 1

Version 01 #1600-116-006

Document Approval Sheet

Document No: 1600-116-006 Version 01

Document Name: DFACTT Data Fusion Tools & Techniques
Report on Advanced Technologies Volume 1

Approvals	**Signature**	**Date**
Author	L. Dumouchel	12 April 1995
Author	C. Crabb	13 Apr 1995
Author/Project Manager	A. George	12 Apr 95
Manager, Land Systems	J. Perrin	12 April 95

P498259.PDF [Page: 6 of 82]

Document Revision History

Revision	Reason For Change	Origin Date
01	Original document issued.	24 March 1995

P498259.PDF [Page: 7 of 82]

TABLE OF CONTENTS

1. INTRODUCTION 1

2. TECHNOLOGY DESCRIPTIONS 2
2.1 Parallel Processing 2
2.1.1 Introduction 2
2.1.2 Architecture 3
2.1.2.1 SIMD vs MIMD 3
2.1.2.1.1 SIMD 3
2.1.2.1.2 MIMD 4
2.1.2.2 Interconnection Architectures 5
2.1.3 Processors and Systems 7
2.1.4 Programming For Parallel Systems 8
2.1.4.1 Concurrent Object-Oriented Programming Languages. 8
2.1.4.2 Automatic Parallelization Techniques. 8
2.1.5 Operating Systems 9
2.1.6 Applications 9
2.2 Acousto-Optic Processors 11
2.2.1 Technology Description 11
2.2.1.1 Background 11
2.2.1.2 Acousto-Optic Processors 11
2.2.2 Acousto-Optic Applications 13
2.2.3 Spread Spectrum Signal Acquisition and Processing 13
2.3 Neural Networks 15
2.3.1 Technology Description 15
2.3.1.1 Background 15
2.3.1.2 Neural Networks 15
2.3.1.3 Network Application Models 19
2.3.2 Neural Networks Applications 22
2.3.2.1 Chemical Engineering 22
2.3.2.2 Diagnostic Expert Nets 22
2.4 Expert Systems 24
2.4.1 Technology Description 24
2.4.1.1 Background 24

P498259.PDF [Page: 8 of 82]

2.4.1.2 Expert System Architecture 28
2.4.1.3 Expert System Components 30
2.4.2 Expert System Applications 32
2.5 Object Database Management Systems 34
2.5.1 Technology Description 34
2.5.1.1 Background 34
2.5.1.2 Object Oriented Database Management System 36
2.5.2 ODBMS Applications 36
2.5.2.1 ORION 36
2.5.2.2 GEMSTONE 36
2.5.2.3 IDB Object Database 37
2.5.2.4 VERSANT ODBMS 37
2.6 Computer Security 40
2.6.1 Technology Description 40
2.6.1.1 Background 40
2.6.1.2 Canadian Trusted Computer Product Evaluation Criteria . . 41
2.6.1.3 Computer Security Policy Models 42
2.6.1.4 Security Certification and Accreditation 42
2.6.2 Trusted Operating Systems 43

3. APPLICATION TO LAND ELECTRONIC WARFARE 44
3.1 Problem Areas 44
3.2 Technology Solutions 46

4. RESEARCH AREAS 50
4.1 Sensor Technologies 50
4.2 Analysis and Command and Control 50
4.2.1 Information Processing Technologies 50
4.2.1.1 Parallel Processing 50
4.2.1.2 Expert Systems 51
4.2.1.3 Object Database Management Systems (ODBMS) 52
4.3 Computer Security 54

5. Research Requirements 56
5.1 Sensor Technology 56
5.1.1 Acousto-Optic Processors 56

P498259.PDF [Page: 9 of 82]

5.1.1.1 Research 56
5.1.1.2 Risks 56
5.1.1.3 Benefits 56
5.1.1.4 Costs and Risk - Benefit Ratios 57
5.1.2 Neural Networks 58
5.1.2.1 Research 58
5.1.2.2 Risks 58
5.1.2.3 Benefits 58
5.1.2.4 Costs and Risk - Benefit Ratios 59
5.2 Analysis and Command and Control 60
5.2.1 Parallel Processing 60
5.2.1.1 Research 60
5.2.1.2 Risks 60
5.2.1.3 Benefits 60
5.2.1.4 Costs and Risk - Benefit Ratios 60
5.2.2 Expert systems 61
5.2.2.1 Research 61
5.2.2.2 Risks 61
5.2.2.3 Benefits 62
5.2.2.4 Costs and Risk - Benefit Ratios 62
5.2.3 Object Database Management Systems 64
5.2.3.1 Research 64
5.2.3.2 Risks 64
5.2.3.3 Benefits 65
5.2.3.4 Costs and Risk - Benefit Ratios 65
5.3 Security 66
5.3.1 Research 66
5.3.2 Risks 66
5.3.3 Benefits 67
5.3.4 Costs and Risk - Benefit Ratios 67

6. Recommendations 68

P498259.PDF [Page: 10 of 82]

LIST OF FIGURES

Figure 1 Interconnection Architectures 6
Figure 2 Basic Construct of an Acousto-Optic Processor 12
Figure 3 Information Processing in a Neurode 16
Figure 4 Typical Neural Network 18
Figure 5 Neural Network Learning Process 20
Figure 6 Recurrent Backpropagation Network 21
Figure 7 Architecture of a Generic Expert System 29
Figure 8 Example of a rule 30
Figure 9 Expert System Components 31
Figure 10 Data Management Generations 34
Figure 11 Technologies of an Advanced EW Process 49
Figure 12 Risk vs Benefit 56

DFACTT Data Fusion Tools & Techniques
Report on Advanced Technologies
Volume 1

Version 01 #1600-116-006

1. INTRODUCTION

This paper, developed under contract number W7714-2-9656/01-QC, discusses several emerging technologies which may have application in the field of Land Electronic Warfare. The technologies discussed are:

1. Parallel Processing;
2. Acousto-Optic Processing;
3. Neural Networks;
4. Expert Systems;
5. Object-Oriented Databases; and
6. Computer Security.

Section 2 provides brief descriptions of the technologies. It discusses some examples of ways in which the technologies have been used and shows both the strengths and the weaknesses of each technology. Appendices to this paper provide additional examples of some of the technologies. Section 3 discusses ways in which the technologies can be combined such that weaknesses of one technology are balanced by the strengths of others in support of Land Electronic Warfare. Section 4 describes areas where further research could be effectively undertaken to maximize the potential of these technologies in the development of an Advanced Land electronic Warfare capability. Section 5 provides rough estimates of the risks, benefits and the personnel effort which would have to be applied to achieve the desired capability and discusses where the maximum benefit is likely to be achieved. Section 6 provides recommendations with respect to prioritizing the research.

2. TECHNOLOGY DESCRIPTIONS

2.1 Parallel Processing

2.1.1 Introduction

Parallel processing can be defined, for our purposes, as a machine in which two or more processors can simultaneously process different parts of a task which has been loaded as a single program [9]. Parallel processing has the potential to achieve performances exceeding that available on any single-processor architecture. Parallel processing is only now breaking into the mainstream, with the support of vendors such as Sun Microsystems (Sparcstation 10/Model 52), Oracle (running on NCUBE hypercubes), IBM (reselling Parallan Computer superservers), and Microsoft (Windows NT multiprocessing support) [2].

The main force behind the development of parallel processing has been the constantly increasing need for performance. For the average computer user, the prospect of parallel computing will provide a better price-performance ratio [9]. Instead of a small number of expensive processors (e.g., Cray), a large number of mass-produced processors can be used to achieve the desired performance at a much lower cost.

Parallel computers have some drawbacks. In general, they do not have the I/O channels, different levels of memory, multiple language compilers, and the programming environments required for the development of applications to run on these machines. To overcome these problems, many users have approached the parallel processing philosophy through the use of small numbers of relatively large machines. This approach offers less in the price-performance category, but provides the support tools and environments to produce the required applications. This parallel design has the ability to spread a task across the entire system, or to allocate each computer to a different task [9].

Another rationale for parallel processing is based on the premise that the semiconductor industry is rapidly approaching the limits of MOS fabrication technology. If you boost clock speeds too far, you have a radar transmitter instead of a computer, and if you make the "wires" too thin, quantum effects will make transistors unpredictable. However, some experts argue that technology advances will overcome these difficulties. Regardless, parallel processing can provide users with large increases in performance, without waiting for the technological advances [2].

2.1.2 Architecture

The design of a parallel architecture primarily concerns the type, number, and configuration of the node processors. The classification of a parallel system in terms of the size and number of its processors is usually referred to as its granularity. In general, systems with many small processors are fine-grained, and those with few large processors are coarse-grained. However, just as important is the memory, the communications, and processor autonomy. Each of these options will be described in the following sections.

2.1.2.1 SIMD vs MIMD

Traditionally, parallel processing architectures have been divided along two lines: single instruction/multiple data (SIMD), and multiple instruction/multiple data (MIMD). SIMD computers execute the same instruction on many data values simultaneously, also known as data-level parallelism. An example might be an array processor which calculates a large two-dimensional matrix of values in parallel by applying the same operation to every element. MIMD involves connecting a number of processors that run different programs or parts of a program on different sets of data, also known as control-level parallelism. Communication between the processors is essential to the success of the MIMD architecture [2]. The communication strategy used provides a further classification: shared-memory and distributed-memory machines.

2.1.2.1.1 SIMD

The SIMD architecture provides a system designed for data-level parallelism; That is, when a single instruction causes the processing of many bits of data. This type of parallelism exists in vector processing machines, where each instruction operates on a vector, array, image, or other large data structure. A fine-grained machine with massive data-level parallelism may have a very small number of single-bit operations, but very complex data movement.

SIMD machines are well suited for speeding up the matrix calculations used in finite-element analysis, seismic modelling, atmospheric modelling, or image processing. They are less suitable for general computing tasks in which the data is not normally expressed as a large uniform array [2]. In terms of general computing, SIMD architectures may not be the best solution; however, for specific parallel computing purposes, they are often the most cost-effective. The processors in a SIMD machine do not need the broad range of capabilities that

P498259.PDF [Page: 14 of 82]

are required by processors in a MIMD machine [2].

2.1.2.1.2 MIMD

The MIMD architecture provides a system designed more for control-level parallelism; That is: when each processor independently executes a sequence of instructions taken from a program.

Shared Memory Architecture. In a shared-memory MIMD computer, all the processors access a common memory, usually over a shared high-speed bus. In theory, this makes communication between processes reasonably simple. However, when you have many CPUs attempting to access the same memory, the bus can be overloaded and cause major slowdowns. The solution to this problem is to add a cache to each processor, although this approach can produce cache consistency problems. There may be several versions of the same variable with different values in different caches. These two problems severely limit the number of processors you can put into a bus-based shared-memory system (i.e., its scalability) [2]. A shared memory system may also be implemented using a switching network, such as a crossbar switch or an omega network. These both require a large number of switches as the number of processors increases (i.e., n^2 and $n\log_2 n$ switches respectively) [3].

The shared-memory architecture is the most widely used, since it is the easiest to program. This results in there being a lot of software and environment support, and also a lower cost. However, its limited scalability implies that its maximum performance is still based on the speed of its CPUs, and not on the number. Also, the cache-coherency problems leads to a certain level of overhead to maintain consistency.

Distributed Memory. In a distributed-memory MIMD machine, each CPU has its own local memory (together they are commonly referred to as a node). Since each node's memory is private, data must be passed between the nodes over a communication network. This type of system is also known as a multicomputer, or a message-passing architecture. The performance of this type of architecture is tied to both the processors and the communication network. For example, adding more processors may not improve the performance if the system is spending most of its time communicating while some of its processors are idle [2].

Scalability is the greatest advantage of the distributed-memory over the shared-memory architecture. In a shared-memory system, adding processors can hamper the performance due

P498259.PDF [Page: 15 of 82]

to the limitations of the bus. The performance of a distributed-memory system can be almost linearly increased by increasing the processors and the corresponding communication bandwidth [2]. However, a distributed architecture is more difficult to program than a shared-memory architecture, and this has meant that for programmers there is little software support. The limited demand has also meant higher costs for those users requiring the performance of hundreds or thousands of processors.

	Distributed-Memory	Shared-Memory
Pros	linear scalability; fine-grained parallelism	easy to program; plenty of software; lower cost
Cons	difficult to program; relatively expensive; little software support	limited scalability; cache-coherency issues; coarse-grained parallelism

Table 1: Comparison of memory architectures [2].

2.1.2.2 Interconnection Architectures

There are a multitude of interconnection architectures for parallel processing systems, and new ones are always being designed. Some common types include systolic arrays, hypercubes, crossbar switches, or omega networks. Figure Figure 1 shows the physical structure of these architectures. There are combined processor and memory nodes (N), switches (S), processors (CPU), and memory (MEM). The lines represent communication links. Those that are linked through nodes, must communicate by traversing through the nodes as required. When linked through switches, these switches must be controlled to provide direct access between the processor and memory.

P498259.PDF [Page: 16 of 82]

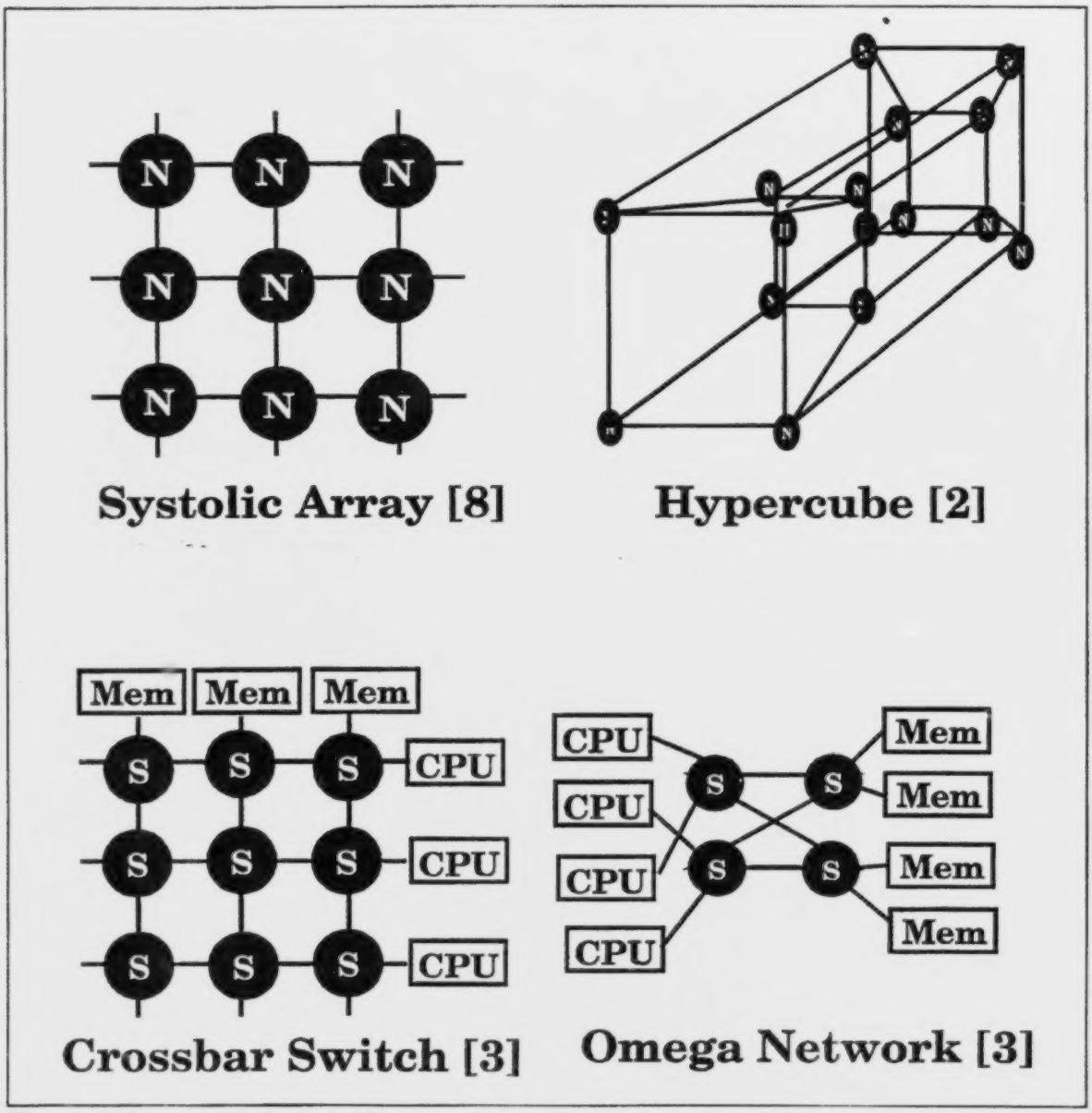

Figure 1 Interconnection Architectures

2.1.3 Processors and Systems

The choice of basic VLSI components is a primary consideration in the design of a parallel system. Designers have the choice of obtaining custom-built chips, or using off-the-shelf chip designs. The established off-the-shelf chips have the advantage of a larger investment in design and technology. However, custom chips can be tailored to suit the requirements of a particular architecture [9]. The major 32-bit microprocessor chip manufacturers include Motorola, National Semiconductor, and Intel. However, there are also 32-bit microprocessors which have been designed from the beginning with parallel computing in mind, such as the Inmos transputer [9].

Parallel Systems. A number of companies provide shared-memory multiprocessor systems aimed at the network-server or workstation market. This has been spurred on by the introduction of Windows NT's support for multiple processors. Contrasted to this, there are also large-scale multiprocessor systems based on special purpose hardware. Available systems are listed below and briefly described in Appendix A.

Company	System
ALR	ProVEISA
Compaq	ProLiant 2000
Vtech	Platinum
Parsytec	GigaCube
Thinking Machines	Connection Machine
Performance Computer Industries (PCI)	CS-2
NCUBE	NC3
Concurrent Computer Corporation	Maxion
Motorola	Series 900
MicroWay	MicroWay Quadputer

P498259.PDF [Page: 18 of 82]

2.1.4 Programming For Parallel Systems

Programming parallel computers is more difficult than sequential programming, with distributed -memory machines being more difficult than shared-memory machines. A shared-memory system will let you use the same global data structures, rather than requiring that you partition your data between distributed memories [2]. The implementation of a multiprocessor application requires building the concurrency, the accompanying coordination, and good load balance so that processors do not sit idle for lack of work. Also, in a shared-memory system, data locality must be well designed so that individual processors do not waste time waiting for memory references to complete [5]. Important consideration must be given to avoiding deadlock, livelock, and race conditions.

In terms of shared-memory systems, the Linda programming model provides data packets (tuples) which are shared in an abstract memory area [2, 3]. The abstraction allows this model to be used in a distributed shared-memory system, where the memory is physically separated but abstractions are available to allow programming as if it were a shared-memory system. The Opal system from Torque Computer includes a portable C implementation of Linda [2].

2.1.4.1 Concurrent Object-Oriented Programming Languages.

Some concurrent object-oriented languages are extensions to existing languages, like Smalltalk and C++, while others were designed from the beginning with parallel processing in mind. There are many such languages such as Argus, Concurrent Smalltalk, POOL-T, Presto, Orca, Mentat, and COOL [1, 3, 5, 7]. However, not all are truly object-oriented due to limited inheritance capabilities.

2.1.4.2 Automatic Parallelization Techniques.

Significant research has been devoted to developing compilation techniques that automatically transform programs to execute on parallel computers. These transformations can reduce task scheduling overhead, increase the program's available parallelism, and improve cache performance. In fully automatic compiler-based approaches the compiler is responsible for performing dependence analysis, and finding and exploiting opportunities for parallelism (automatic parallelization). Unfortunately, compilers are best at finding fine-grain and loop-level parallelism, and not good at detecting large-grain parallelism [7]. Message-passing

P498259.PDF [Page: 19 of 82]

MIMD architectures, which require medium to coarse grain parallelism to operate efficiently, are inappropriate for this approach [7].

2.1.5 Operating Systems

Unix is a multitasking operating system that is widely used for multiprocessing. For example, multiprocessing extensions have been added by Corollary and The Santa Cruz Operation in SCO Unix with the MPX operating system. Posix.4 defines extensions to Posix.1 that include many features important for multiprocessing. LynxOS from Lynx Real-Time Systems and TC/IX from Control Data are both Posix compliant, as is Motorola's Unix [2, 12].

2.1.6 Applications

Many applications are being researched or have already been built to take advantage of parallel systems. The following is a brief description of some of these applications.

General Motors has been active in this area; in vehicle aerodynamics, they have used a parallel programming approach to simulate air flow around a car body model. The approach involves solving a number of linear equations equal to the number of segments into which the car model is segmented, e.g., 1000 to 3000 [9]. They are also studying engine combustion efficiency through a package called KIVA, for three-dimensional modelling of the mix of fuel, air, and vapour through an engine. Solutions require solving 20 equations in each of over 9000 cells, for possibly 2000 time-steps, and include a graphical representation of the results [9]. Other examples are to reduce engine emissions, to study the chemistry of fuel-air ignition mixtures, and to model, find, and eliminate the sources of noise in the passenger compartments of their vehicles [9].

The Connection Machine from Thinking Machines has been used in numerous applications, such as demonstrating database-retrieval capabilities. A "Words" demonstration provides a keyword search, and also a form of pattern matching to retrieve a particular document(s). As the source documents are stored, keywords and phrases are chosen from the document and stored in a "content kernel". Each of these kernels are distributed across the nodes of the Connection Machine. The user enters a request for a document search, the system identifies the keywords and phrases, and each node searches its own kernels for any matches [9]. In a similar application, it has also been used in an effort to improve the performance of speech-to-speech translations. Each node on the Connection Machine (up to 65,536) is loaded with

P498259 PDF [Page: 20 of 82]

a few sentences. A sentence is entered, and each node searches its own set of sentences for similar sentences. The most appropriate would have the required words (e.g., the subject) changed [10].

There is some research currently in the development of an automated dictation system, providing speech recognition and word-processing capabilities. This would involve identifying the speech sounds, then the breaks between each succession of sounds, and matching each succession with a word known to the system. Once the words are identified, each sentence must be determined and analyzed to ensure proper punctuation, and that the correct words have been chosen. Estimates of 100 mips performance are required to reach the objective of 150 words per minute. The Flagship project in Britain, and the Kurzweil Voice Writer project are both researching the development of such a system [9].

RSIC in the USA is building a parallel data compression/decompression system for the VME bus. This is designed to reduce the problem of the transmitting and storing of the large amounts of data often required in computationally intensive applications. The implementation uses a systolic pipe architecture applied to over 4000 VLSI processors on one board [2].

The CNAPS (Connected Network of Adaptive Processors System), built by Adaptive Solutions, contains 256 special-purpose VLSI processors and a 68030 control processor. The company employs the system in pattern-recognition problems such as voice recognition [2].

2.2 Acousto-Optic Processors

2.2.1 Technology Description

2.2.1.1 Background

Radio frequency spectral analysis defines a subset of a class of problems which are difficult to perform using digital techniques. This is due to the high sampling rates required and the resulting computationally intense calculations which must be performed. Specific examples of problems which fall into this category include Total Power Radiometry, Interference Rejection Filtering and Spread Spectrum Signal Correlation.

In 1932, P. Debye and F.W. Sears of the department of physics at the Massachusetts Institute of Technology published a paper entitled "On the Scattering of Light by Supersonic Waves" [13]. In their paper, they describe an experiment where supersonic waves are induced to travel longitudinally in a liquid in a transparent rectangular trough. When a fine beam of light is presented incident at an angle to the direction of motion of the supersonic waves, the light is diffracted and numerous spectral lines appear on a detecting screen. They further discovered that the diffracting angle changed as a function of the frequency of the supersonic signal. They theorized (though it is not stated as such) that harmonics of the supersonic signal produce additional spectral lines.

Since the paper was originally published, the results of the experiment have been extensively studied and refined. Various crystalline substances have been employed to produce cells (a sub-class of which are called Bragg cells) which exhibit the properties noted above. Different cells have different center frequencies and bandwidths at which they operate optimally.

2.2.1.2 Acousto-Optic Processors

An Acousto-Optic processors is, essentially, an analog as opposed to a digital device. Figure 2 shows a simple schematic of the basic construction of such a device. The light passes through a slit and shines on the Bragg Cell. With no signal source present, the light is not diffracted and appears as a single band on the detector. When a signal is applied to the transducer, supersonic waves are induced in the crystal of the Bragg Cell causing a diffraction of the light.

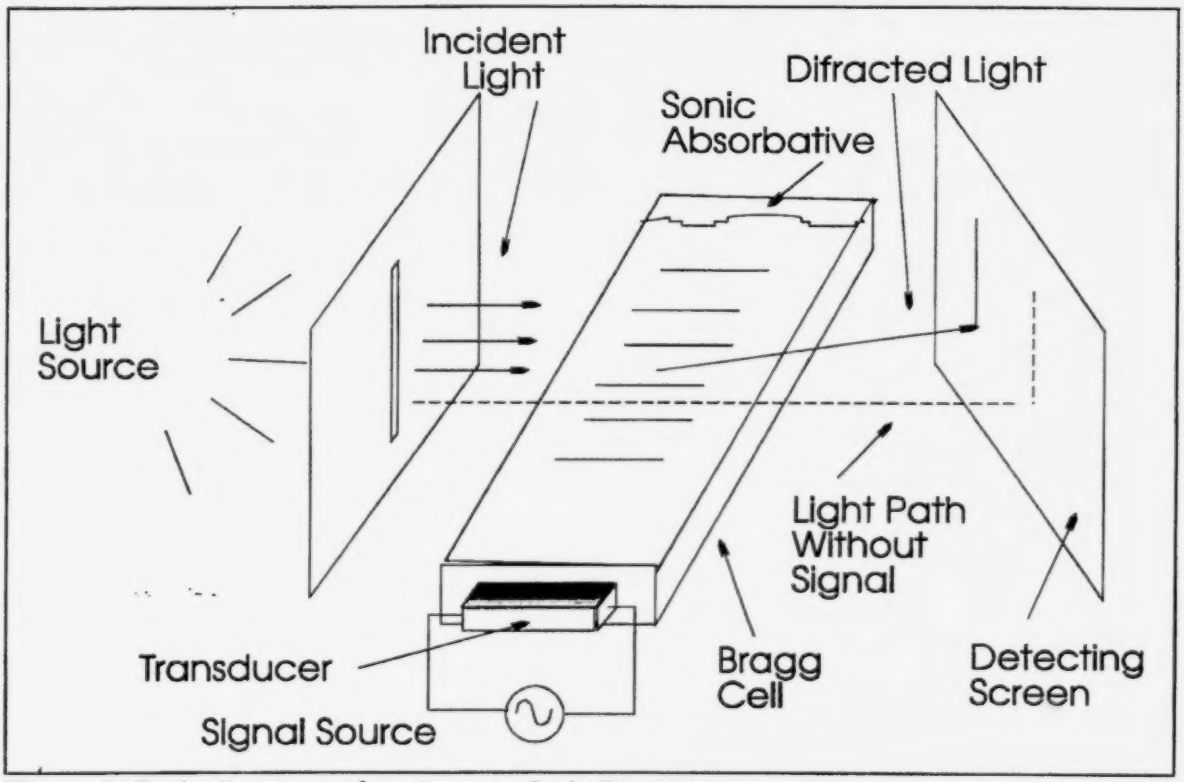

Figure 2 Basic Construct of an Acousto-Optic Processor

Use of an appropriate detectors such as photo-optic diodes connected to charge coupled devices permit the frequency calculation to be performed in a relatively straight forward manner. Terry M. Turpin in his 1981 paper "Spectrum Analysis Using Optical Processing" [14] suggests that reductions in data rates of the order of 10^{-3} are readily achievable using acousto-optic techniques.

2.2.2 Acousto-Optic Applications

In the last 20 years, acousto-optic component technologies have moved from the research lab to commercial availability. This has meant that components are now being used to develop processors capable of tackling some truly difficult problems.

Following are three examples of problem solutions using acousto-optic techniques.

2.2.3 Spread Spectrum Signal Acquisition and Processing

N. Brouseau and J.W.A. Salt, in a technical note published in December 1993 [15], describe a system which uses an expanded HeNe laser in a Mach-Zehnder configuration to produce a correlation peak with respect to two RF signals. That design provides for a wide dynamic signal range suitable for spread spectrum signal acquisition and correlation or alternatively, direction finding through the correlation of the time of arrival of a specific signal at two widely spaced antennae. Their prototype is mountable on a 19" rack making its use for field operations a real possibility.

The model developed for the prototype described in the paper does not perform the correlation in real time though incorporation of the post-processing correlation algorithms into hardware could probably achieve real-time or near real-time results.

Radar-Frequency Estimation using an Acousto-Optic Spectrum Analyzer

In [16] Guy Farley presents a statistical model for calculating the Cramer-Rao bound for the estimation of the carrier frequency of radar signals detected using an Acousto-Optic Spectrum Analyzer. He also presents an algorithm for sidelobe rejection. He postulates that Radar Electronic Support Measures Systems will, in the future, be exposed to extremely dense radar environments where simultaneous arrival of different radar pulses will be likely. Current systems tend to make erroneous measurements and mix signals under these conditions.

The Acousto-Optic Spectrum Analyzer (AOSA) together with the statistical model and algorithms presented in this paper support the contention that an AOSA is a suitable candidate technology for managing the RESM problem in a dense threat environment.

P498259.PDF [Page: 24 of 82]

Simultaneous Spectrum Analysis and Direction Finding

Jim P.Y. Lee describes two antenna architectures used to feed the multi-channel bragg cells of an acousoto-optic processor [17]. In one architecture a linear antennae array is used where each element of the array feeds a separate channel of a bragg cell being illuminated by a laser light. The resultant optical pattern can be used to determine both the frequency of various signals and their time of arrival permitting a determination of the direction of the signals (to an error of about 1^0). This arrangement is quite accurate as long as the elevation angle relative to the source is near 0^0 and the required field of view is only one quadrant.

The second arrangement involves a 32 element circular antennae array. It is less subject to elevation errors and it has a 360^0 field of view. Its accuracy, however, is somewhat worse than the linear array.

These three examples demonstrate that acousto-optic processors have the potential to reduce the data flow in a dense emitter environment to more manageable levels. There remains a problem with processing the arrays of data at the near real-time speeds necessary for further reduction of the resultant information flows to a level which humans can process.

2.3 Neural Networks

2.3.1 Technology Description

2.3.1.1 Background

Artificial Intelligence (AI) can be defined as follows:

> *.....behaviour by a machine that, if performed by a human being, would be called intelligent.*

AI problems are generally ill-defined, and the theories proposed are often too complex and involved to be verified by intuitive or formal arguments. To this end, programs are written that are intended to reflect the theories proposed. With this emphasis on writing programs, it is important that AI researchers have a wide library of programming tools available. Some of the more commonly used tools for programming AI theories are: discrimination nets, agendas, deduction, data dependencies, backtracking, and neural networks. The rest of this section will focus on neural networks.

2.3.1.2 Neural Networks

A neural network is comprised of processing elements organized in a network structure. These processing elements are called neurodes. Each neurode receives input, processes the input and delivers a single output. The input can be raw data or the output from another neurode. Figure Figure 3 shows this process.

Information Processing. Neural networks process information in a dynamic, self-adjusting way typical of living systems. The elements described in Figure Figure 3 participate in the processing of the information.

Inputs. Each input corresponds to a specific attribute. The *value* of the attribute is the input to the neurode or network. Several kinds of data can be used as input to a neurode/network.

Outputs. The output of the network is the solution to the problem. The purpose of the network is to calculate the value of the output.

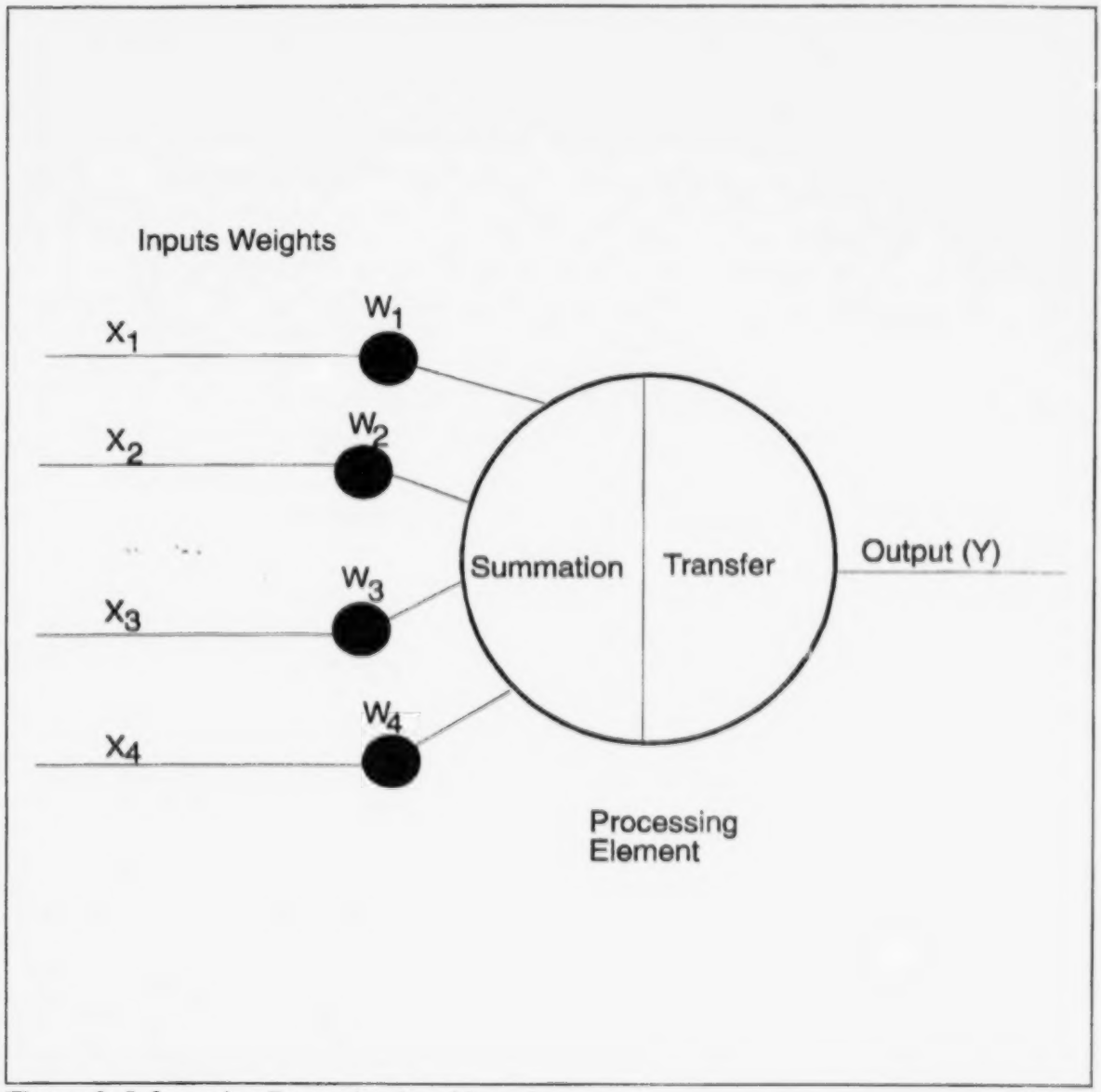

Figure 3 Information Processing in a Neurode

P498259.PDF [Page: 27 of 82]

Weights. This is a key element of the network. Weights express the *relative strength* of the input data or of the various connections that transfer the data from layer to layer. Weights are crucial. It is through repeated adjustments of the weights that a network "learns". This process will be described later in this section.

Summation function. The summation function finds the weighted average of the inputs to a processing element. A summation function multiplies each input value by its weight, and totals them together for a weighted sum.

Transfer function. This function expresses the relationship between the internal activation level (the result of the summation function) and the output of a neurode. The relationship may be linear or nonlinear and there are several types. The selection of a specific function determines the operation of the network. The purpose of this transformation is to modify the output levels to more reasonable values, normally between zero and one. This transformation is done before the outputs reach the next level. Without such transformation, the value of the output could be very large, especially if many layers are involved.

In general, a neural network is made up of any number of layers with any number of neurodes in each layer. However, most neural networks in existence today have three layers. The first layer is designated the input layer, the last the output layer and those in between the hidden layers. Figure Figure 4 illustrates a typical neural network.

Learning (or training). An important part of working with neural networks is to understand how a neural network learns or is trained to perform its functions. A neural network learns from running a set of training data through it and adjusting the weights to produce the desired output for each of the inputs. The usual process of learning or training involves three tasks as shown in Figure Figure 5:

a. compute outputs;

b. compare outputs with desired answers; and

c. adjust the weights and repeat the process.

The learning process starts by setting up the weights randomly. The difference between the actual output and the desired output is called delta. The objective is to minimize delta or

P498259.PDF [Page: 28 of 82]

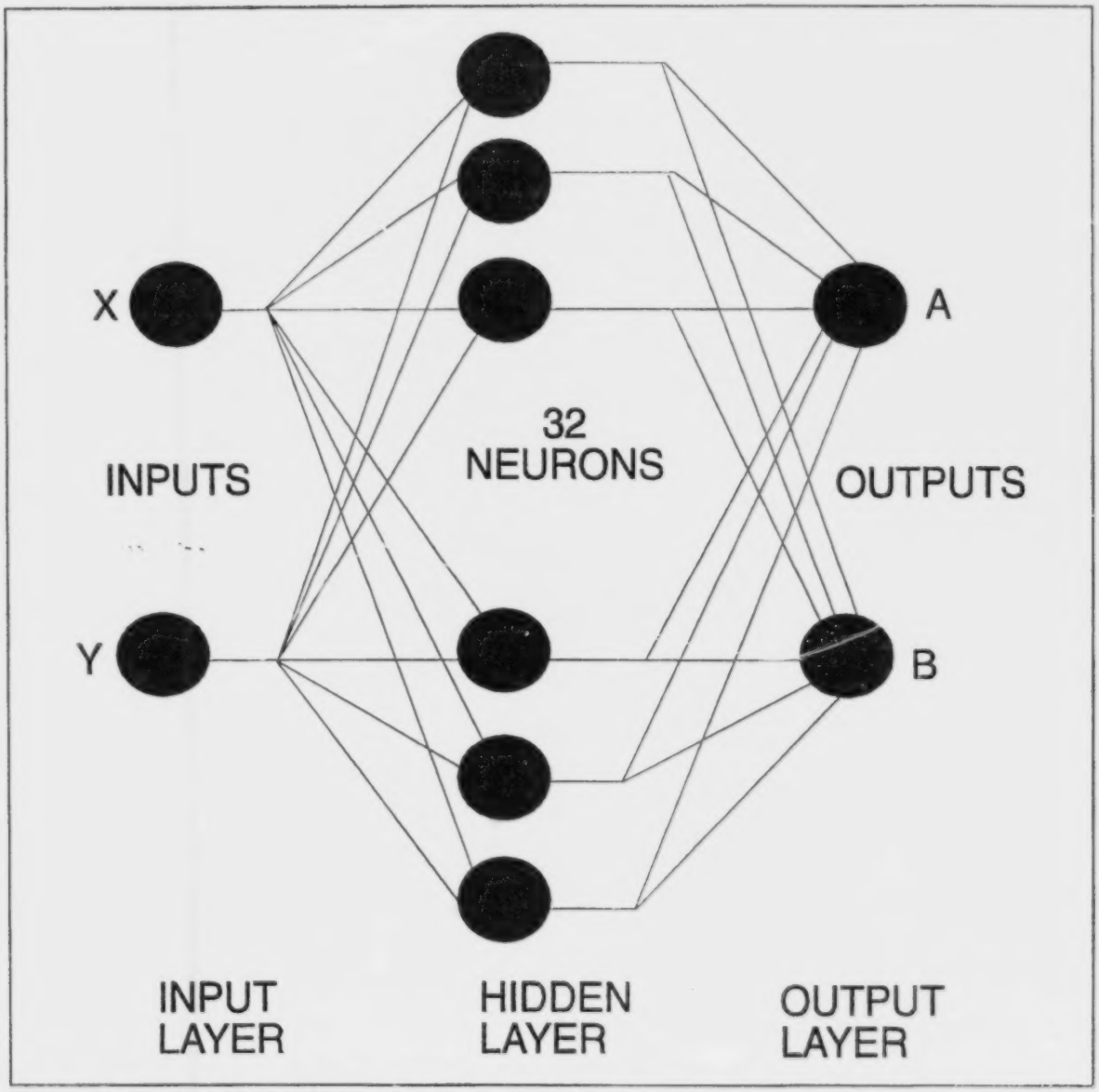

Figure 4 Typical Neural Network

P498259.PDF [Page: 29 of 82]

reduce it to zero.

Supervised Learning. Supervised learning uses a set of inputs for which the appropriate (desired) output is known. The difference between the desired and actual output is used to adjust the weights of the neural networks. Backpropagation is one example of this type of learning, and will be described later in this section.

Unsupervised Learning. In unsupervised learning, only input stimuli are shown to the network. The network organizes itself internally (the weights are adjusted) so that each hidden processing element responds to a different set of input stimuli. No knowledge of what the output should be is provided. A human must analyze the final output to assign meaning and to determine the usefulness of the results. Examples of this type of learning include the Adaptive Response Theory (ART) and Kohonen self-organizing feature maps.

2.3.1.3 Network Application Models

There are three major neural networks application models: backpropagation, recurrent networks and recurrent backpropagation. In the past few years, the notion of combining the delta rule with other training techniques to make hybrid, hierarchical systems has gained in popularity.

Backpropagation. Backpropagation networks are the most used networks by far, since they are one of the few networks with relatively simple mathematical proofs that they can achieve some specific results. They also use a supervised training technique, where the network designer dictates the expected results, and against which the network performance is measured.

Recurrent Networks. This general category includes any network design in which the activity must pass through the network more than once before a response is produced. Activity flows around and around until an equilibrium is achieved. At that time the output pattern represents the system's response to the input stimuli.

Recurrent Backpropagation. Recurrent networks are constructed by adding recurrent neurodes to backpropagation models. The recurrent neurodes (black in Figure Figure 6) represent the output of their respective neurodes in the previous iteration. The recurrent neurodes provide each input stimulus pattern with a temporal context. Therefore, a recurrent

P498259.PDF [Page: 30 of 82]

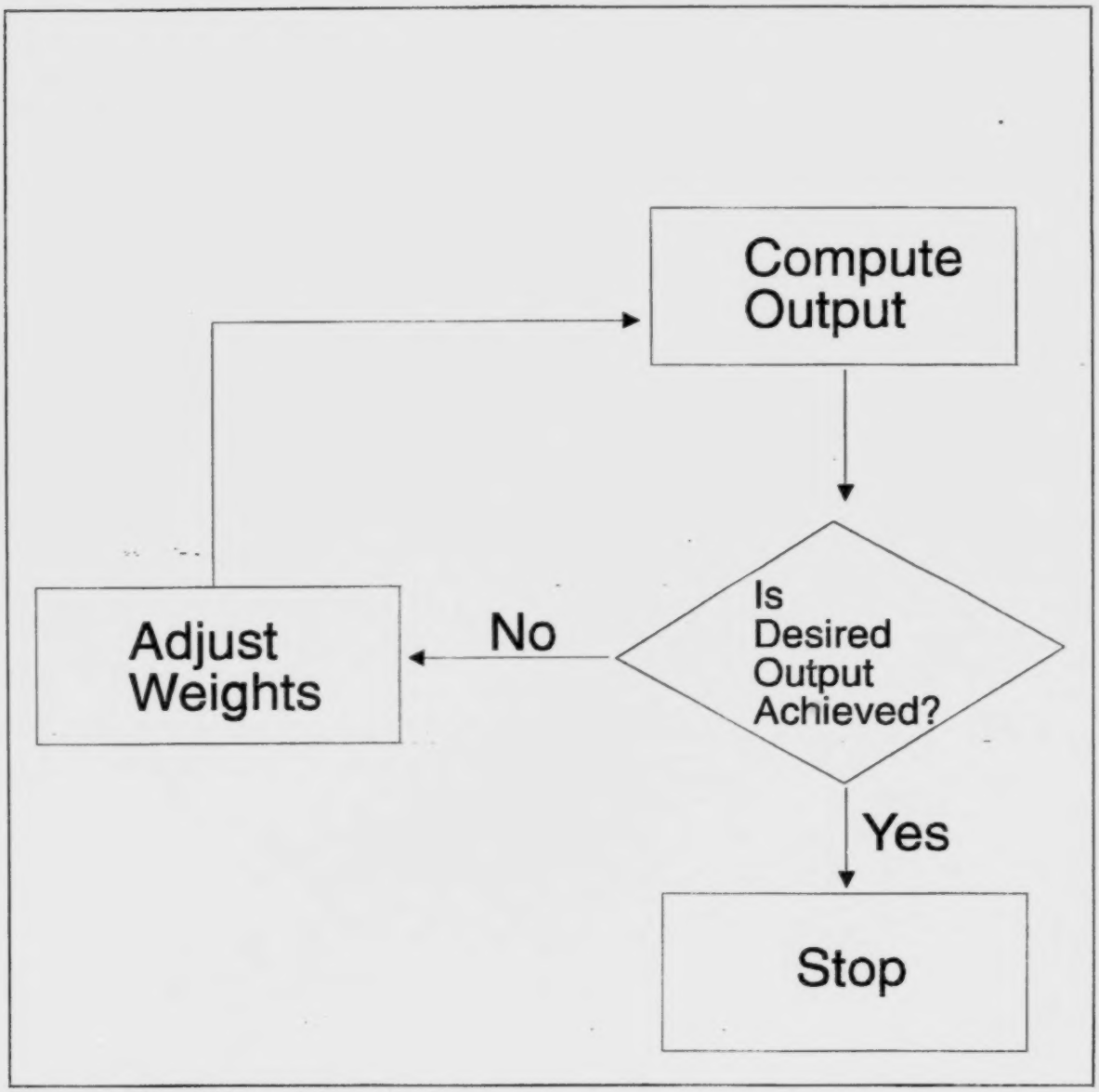

Figure 5 Neural Network Learning Process

P498259.PDF [Page: 31 of 82]

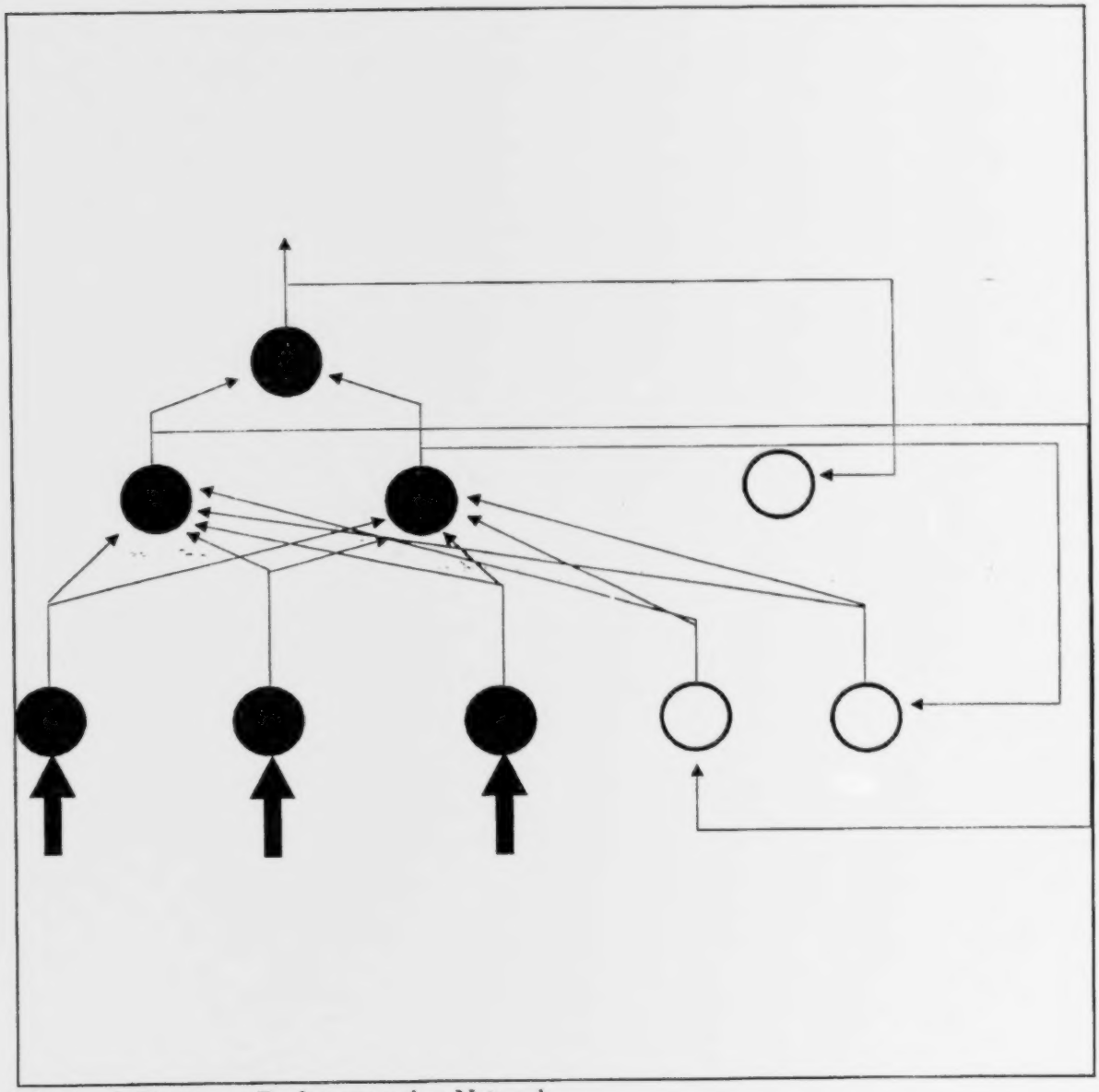

Figure 6 Recurrent Backpropagation Network

P498259.PDF [Page: 32 of 82]

backpropagation network can learn differences in pattern sequences instead of learning simple static patterns.

2.3.2 Neural Networks Applications

The past five years have demonstrated that neural networks have grown from just ideas in the minds of their inventors to successfully engineered systems. Nearly all applications reported have been built, and some are in their second or third generation of development. The following sections outline some current applications. These and others are also described in Appendix B.

2.3.2.1 Chemical Engineering

Neural network technology is very well suited for solving problems in the chemical industry, owing to its wide range of applicability and its ability to learn complex, nonlinear relationships which includes noisy or less precise information. Many chemical reactions and processes involve a mix of heat transfer, fluid mechanics and catalytic and kinetic phenomena which results in nonlinear coupled algebraic or differential equations. Engineering models solving such sets of equations usually can explain only the fundamental phenomena such as heat transfer. However, they can not include several process-related influences such as different machines, seasons, humidity and other process noises. Neural networks can learn complex nonlinear relationships even when the input information is noisy or less precise.

Chemical Kinetics Development. Catalytic chemical reactions are generally described by nonlinear reaction-rate expressions. These expressions are useful in designing a continuous reactor system or modifying a batch reactor system. Nonlinear reaction-rate expressions frequently contain more unknown parameters than can be evaluated by classical chemical-reaction kinetics or nonlinear statistical models. It is not uncommon to see fifteen or more nonlinear parameters in rate models for complex industrial reactions. Neural networks are used to process constant data to obtain a nonlinear kinetic model. The neural network results are simulated to find the effects on the kinetic rate constants. Neural networks have proven more robust than the kinetic power law or statistical quadratic models because of its ability to interpolate nonlinearly the results even if data is missing.

2.3.2.2 Diagnostic Expert Nets

P498259.PDF [Page: 33 of 82]

Neural networks can be used as diagnostic experts in a wide range of applications. They are being used to diagnose everything from automobile engine failure problems to satellite communications faults to medical illnesses. The construction and training approaches are equally varied and present an interesting range of methods and networks.

Testing Digital Circuits. One interesting use of neural networks is to diagnose failures in digital and analog circuit cards. Researchers at the State University of New York (SUNY) in Buffalo have developed a system intended to assist technicians identify faults in circuit boards. This neural network will not only help identify the failure but also direct the technicians to the next appropriate test.

Diagnosing Faults in Satellite Communications. A system called DASA/LARS is currently in operation at Ft Detrick, Md. The system is composed of nine neural networks that take their information from two key sources: a spectrum analyzer and a large database. The complete spectral information is processed by the various networks more or less in parallel. One network looks for unauthorized broadcasts while the remaining networks look for about a dozen other defects in the communication link.

Satellite link planner. Special Operations Forces planners are currently field testing a neural network system developed by SFA Incorporated of Landover, Maryland, to plan UHF satellite communications links. The system ensures a successful communications link with the lowest probability of enemy detection. The neural network system is known as the *viability evaluator* and will enable planners untrained in propagation theory to select optimally from available communications equipment for a given mission. The trained system processes cases with less than 2 percent error. Additional neural network tools are planned to address communications involving HF groundwave and skywave, VHF and UHF line-of-sight.

P498259.PDF [Page: 34 of 82]

2.4 Expert Systems

2.4.1 Technology Description

Since World War II, computer scientists have tried to develop processes and techniques that would allow computers to behave more like human beings. Over the past ten years, a collection of techniques that enables computers to assist people in analyzing difficult problems and making decisions, called knowledge-based expert systems, has proven its value in numerous commercial applications. These systems change the way organizations operate by altering the way people go about solving problems. Expert system technology also helps solve productivity problems by allowing individuals to handle more information in more complex ways. This leads to an increase in the quality and quantity of decisions that are being made. In short, expert systems allow for more information to be gathered, synthesized and put into use more rapidly.

An expert system can be defined as follows [18]:

> *.....an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field.*
>
> *The knowledge of an expert system is made of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in the field. The "heuristics" are mostly private, little-discussed rules of good judgement, rules of plausible reasoning, rules of good guessing that characterize expert-level decision making in the field. The level of performance of an expert system is primarily a function of the size and the quality of the knowledge base it possesses.*

2.4.1.1 Background

P498259.PDF [Page: 35 of 82]

The first expert systems were built in research laboratories by interviewing human experts and attempting to capture these experts' knowledge, hence the term expert system. These systems used programming languages, such as LISP, to represent knowledge and control structures. As experience accumulated, researchers recognized that their systems consisted of two main components with accompanying support facilities: control and reasoning procedures, and domain-specific knowledge. This realization led to systems that isolated inference from knowledge, permitting the reuse of inference procedures (inference engine) with knowledge from other domains. This led to a significant reduction in the time required to implement applications with similar requirements through the creation of *knowledge engineering tools*.

Knowledge engineering tools. These tools allow for the development of expert systems that can be described in terms of one, or at the most a very few consultation paradigms. A *consultation paradigm* is a generic conception of a particular type of problem solving that is common to several different domains. A list of generic types of problem solving is given in Table 3-1. The three better-known consultation paradigms are: *diagnosis/prescription, planning and design*. They are summarized in Table 3-2.

P498259.PDF [Page: 36 of 82]

Table 3-1 - Generic Types of Problem Solving

Interpretation	Inferring situation descriptions from sensor data
Prediction	Inferring likely consequences to given situations
Diagnosis	inferring system malfunction from observables
Planning	Designing actions
Design	Configuring objects under constraint
Monitoring	Comparing observations to plan vulnerabilities
Debugging	Prescribing remedies to malfunctions
Repair	Executing a plan to administer a planned remedy
Instruction	Diagnosing, debugging and repairing student behaviour

P498259.PDF [Page: 37 of 82]

Table 3-2 - Consultation Paradigms

Consultation Characteristics	**Diagnosis/ Prescription**	**Planning**	**Design**
Alternate names	Structured selection, classification	Constraint satisfaction	Discovery
Example	Diagnosis of infectious disease and prescribing drugs	Planning a project, configuring a computer	Designing an integrated circuit
Input	Conditions, answers to questions	Goals, limitations, means	Goals, limitations, means
Knowledge	Expert uses rules-of-thumb and simple calculations	Constraints, templates, heuristics	Constraints, templates, heuristics
Output	Solution heuristics, simple calculations	Planning heuristics, templates, constraints	Discovery heuristics, constraints

2.4.1.2 Expert System Architecture

The architecture of an expert system is shown in Figure Figure 7. The system can be divided into two parts. The knowledge base and the working memory constitute one part of the system. The inference engine and all of the subsystems and interfaces constitute the other part. The knowledge base contains the facts and rules that embody the expert's knowledge. The inference engine contains the inference strategies and controls what an expert uses when manipulating the facts and rules [18].

There are five common strategies used to represent knowledge in an expert system [19]. Each strategy has its advantages and disadvantages. They are discussed in more detail in the following paragraphs.

Semantic Networks. This is the most general representational scheme. It is a collection of objects called *nodes* which are connected by *links*. A node represents an object or descriptor, while a link defines a relationship between objects and descriptors. Flexibility and inheritance are the major advantages of this representation scheme. Inheritance refers to the ability of one node to "inherit" the characteristics of other nodes that are related to it.

Object-Value-Attribute (O-V-A) Triplets. This is another common representation of factual information. An attribute is a general property or characteristic of an object, such as colour, and the value specifies the precise nature, such as red. This representation scheme also allows for the handling of uncertainty. O-V-A triplets can be modified by a certainty factor which represents the confidence one has in a piece of evidence.

P498259.PDF [Page: 39 of 82]

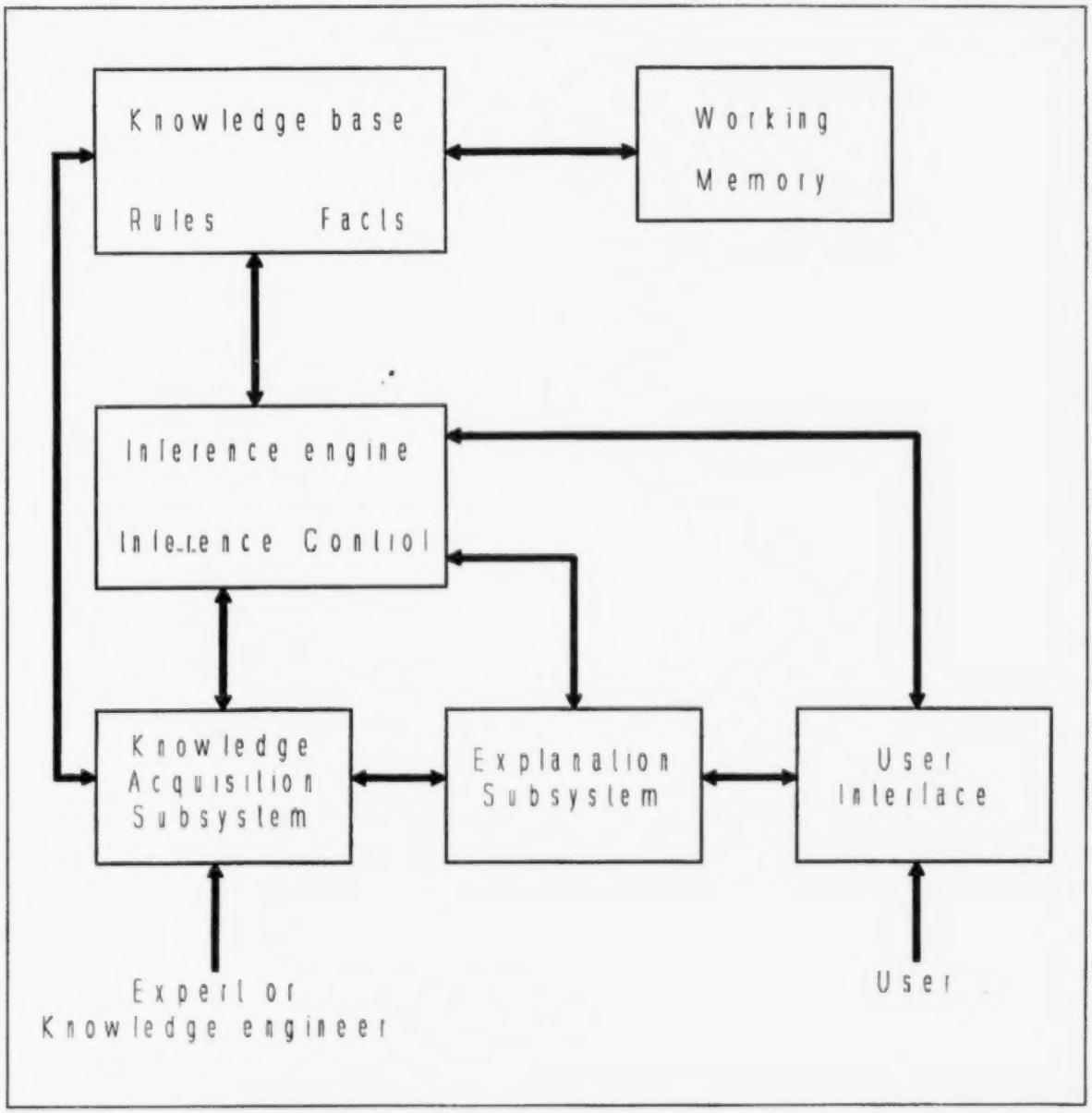

Figure 7 Architecture of a Generic Expert System

Rules. Rules are used to represent relationships, and can be used with O-V-A representations. Rules can be simple or quite complex. Each rule is divided into a *premise* and a *conclusion*, each of which can comprise one or more clauses.

```
Rule:

Premise:      If    correctly dated; and (Clause)
                    signed by originator; and
                    amounts match; and
                    payee identified; and
                    cheque endorsed by payee

Conclusion:   Then  cheque is complete.  (Clause)
```

Figure 8 Example of a rule

Frames. A frame is a description of an object that contains slots for the information associated with the object. Slots, like attributes, may store values. Slots may also contain default values, pointers to other frames, sets of rules, or procedures by which values may be obtained. Facts are either *declarative* (i.e., an assertion that the fact is true), or *procedural* (i.e., a set of instructions that produce a result consistent with a fact).

Logic. There are several logical notations; the two most common forms are propositional logic and predicate calculus. Propositions are statements that are either true or false, and are linked together with connectives, such as AND, OR, and NOT to form compound statements. Propositional logic is concerned with the truthfulness of compound statements. Predicate calculus is an extension of propositional logic based on statements pertaining to objects.

2.4.1.3 Expert System Components

A useful way of thinking about the components of an expert system and the relationships between its parts is to use the dialogue, data and models (DDM) paradigm. There is a *dialogue* between the user and the system, the *data* that supports the system, and the *models* that provide the analysis capability. Figure Figure 9 provides an illustration of the components of an expert system [21],[22].

P498259.PDF [Page: 41 of 82]

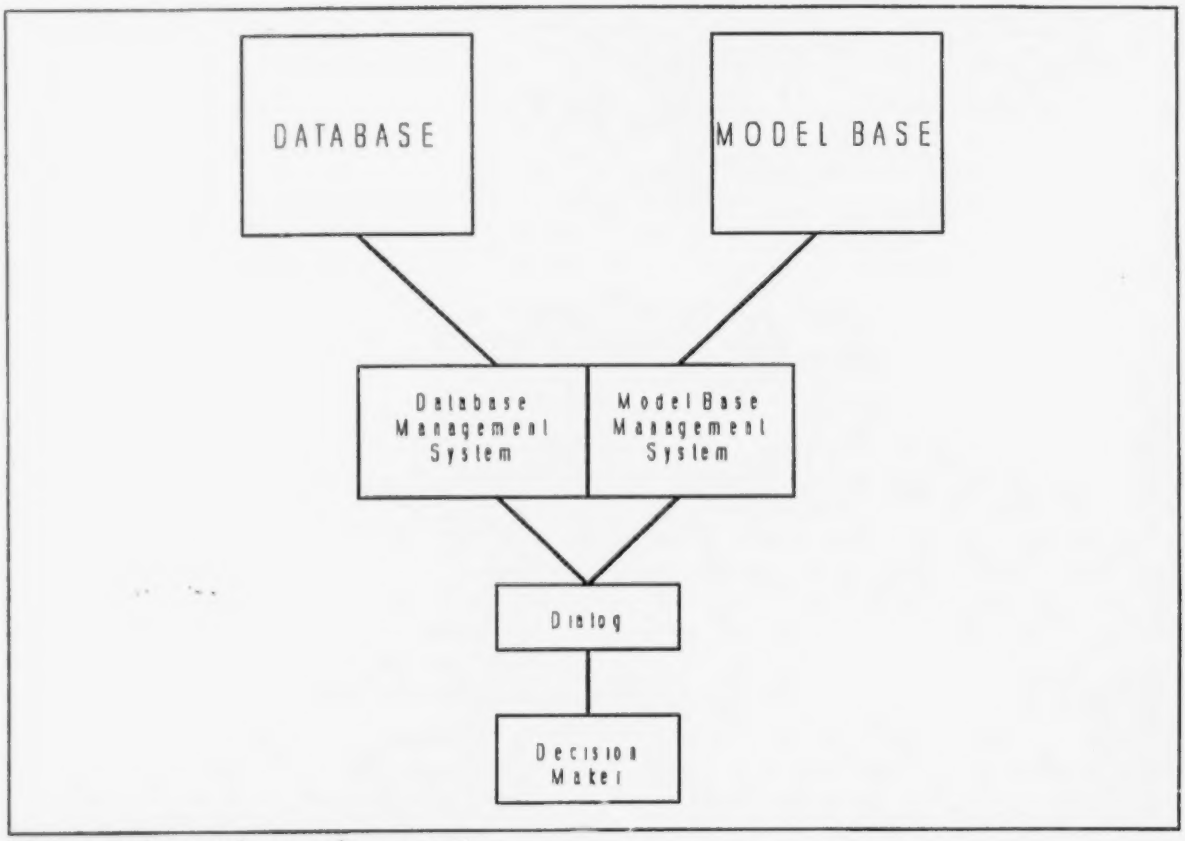

Figure 9 Expert System Components

The Dialogue component. From the user's viewpoint, the dialogue is the interface with the system. As such, much of the power, flexibility and usability characteristics of the expert system are derived from capabilities in the interaction between the system and the user. The elements of the dialogue component are:

a. *the Knowledge Base.* The knowledge base includes what the analysts know about the decision-making and how the expert system works;

b. *the Action Language.* This represents the actions used by the analysts to

P498259.PDF [Page: 42 of 82]

control the expert system; and

c. *the Presentation Language.* This is how output from an expert system is presented.

The combination of the three elements is referred to as "the dialogue style" [20].

The Data component. Data plays an important role in an expert system. It can not be taken for granted. The data can be internal or external to the organization. Internal data refers to the data that is generated by the organization while external data is data from the organization's environment. Care must be taken to ensure that all data is made available to the expert system.

The Model component. Models provide the analysis capability for the expert system. There are many types of models and various ways they can be catalogued. The purpose of a model is either optimization or description. An optimization model seeks to identify points of maximization or minimization while a description model describes the behaviour of a system. Models can be either custom-built (used with only one system) or ready-built (applicable to many systems).

The dialogue, data and models paradigm provides a strong conceptual view for understanding the components and relationships of an expert system. Each is critical if the system is to live up to its decision support potential [20].

2.4.2 Expert System Applications

Recently developed expert systems are still confined to well-defined tasks. They are still not able to reason broadly over a field of expertise. They can not reason from axioms or general theories. They do not learn and, consequently, they are limited to using the specific facts and heuristics they were "taught" by a human expert. They lack common sense, they can not reason by analogy and their effectiveness deteriorates rapidly when problems extend beyond the narrow task they were designed to perform.

On the other hand, these expert systems also do not display biased judgements, nor do they jump to conclusions and then seek to maintain those conclusions in the face of disconfirming evidence. They do not have "bad days"; they always attend to details and they always

P498259.PDF [Page: 43 of 82]

systematically consider all of the possible alternatives. Some of them, equipped with thousands of heuristic rules, are able to perform their specialized tasks better than a human specialist. New concepts and techniques that are being introduced now will undoubtedly result in expert systems that are more powerful and flexible.

Following are two examples of recent expert systems; an overview of further systems is in Appendix C.

TOP SECRET (An Expert in Security Classifications) [23]. TOP SECRET was developed using the EXSYS expert system shell. The knowledge base contains the rules from the classification guidelines that determine how to classify a document. The system is being used to relieve the workload of the people previously assigned to do this complex task who used over 100 classification guides in order to properly classify a given document.

Codecheck (An Expert in Computer Program Assessment) [23]. Codecheck is a rule-based expert system that checks C source code for things such as complexity, formatting and adherence to standards. It identifies those portions of the code that can be simplified. It also evaluates the portability of the source code by comparing it to the numerous standards now existing for C programs, and identifies any code that will not port between DOS, OS/2, UNIX, VMS and Macintosh.

P498259.PDF [Page: 44 of 82]

2.5 Object Database Management Systems

2.5.1 Technology Description

2.5.1.1 Background

Data management technology has evolved through three generations in the last twenty-five years. It is now entering its fourth generation with object database management systems (ODBMS). Figure Figure 10 shows the evolution of data management over the years.

The building of DFACTT in Smalltalk, an Object-Oriented Programming System (OOPS), has led to the evaluation of ODBMS technology to manage the large amounts of data required.

Conventional (relational and prerelational) database management systems have served well the needs of applications they were designed to support, namely business data processing applications. However, a variety of new applications are beginning to emerge that are difficult to implement using relational database management systems. Such applications include:

a. computer-aided design (CAD), engineering (CAE), software engineering (CASE) and manufacturing (CAM) systems;

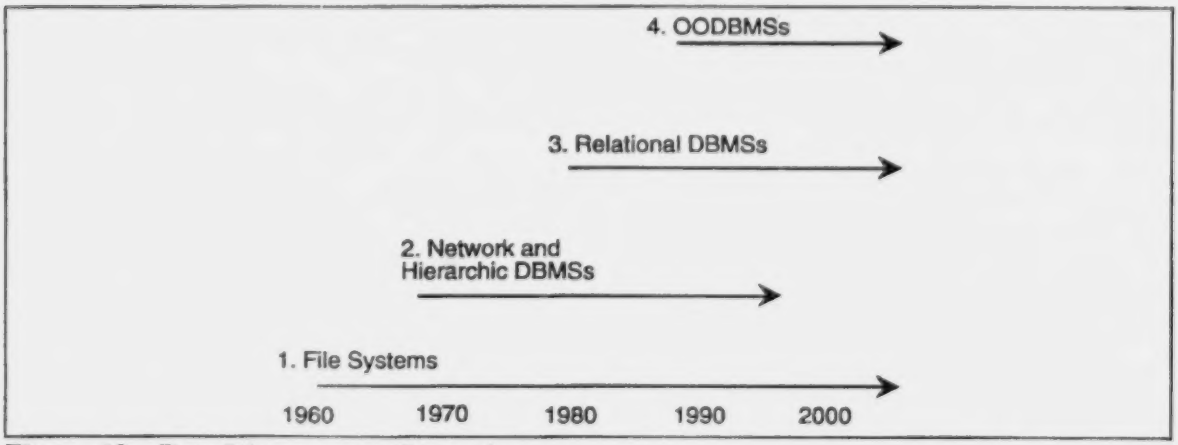

Figure 10 Data Management Generations

P498259.PDF [Page: 45 of 82]

b. knowledge-based systems (expert systems and shells);

c. multimedia systems that manage images, graphics, voice and textual data; and

d. statistical and scientific modelling and analysis programs.

These applications present serious difficulties related to a logical representation of the data and the relationships and constraints on the data. They often require:

a. facilities for modelling and managing complex nested entities;

b. a richer set of data types that includes user-defined data types and long unstructured data such as images and audio;

c. semantic concepts such as generalization and aggregation relationships that are frequently useful; and

d. the concept of temporal evolution of the data.

Some of these applications are highly computer-intensive on a large volume of memory-resident data. Therefore, conventional database management systems can not meet the performance demands of these applications. Also, since some of these applications require long duration, interactive and cooperative transactions, they are not supported well by conventional systems [25].

As data becomes more distributed, varied and complex, posing new data management problems, object database management systems (ODBMS) are cropping up as a more efficient approach to data management. Current computer applications manipulate not only alphanumeric data but also graphics, voice, video, images, documents and text. These complex data types are easier to represent as objects in an object database.

2.5.1.2 Object Oriented Database Management System

An object-oriented database (OODB) is a Database Management System (DBMS) built especially to store, retrieve and manipulate objects rather than simple data types. An ODBMS has been defined in the Object-Oriented Database System Manifesto [26] as having mandatory, optional and open features. These features are described in detail in Appendix D.

2.5.2 ODBMS Applications

For many applications, traditional database models and technologies have proven inadequate and this has led to research and development of database technologies based on the object-oriented paradigm. In this section, some of the recent commercial and prototype object database systems are surveyed. More descriptions and a detail table are provided in Appendix D.

2.5.2.1 ORION

ORION is an object database management system (ODBMS) developed by the Advanced Computer Architecture Program at the Microelectronics and Computer Technology Corporation (MCC) of Austin, Texas [27]. It is currently in use to support an in-house expert system shell called Proteus. The most distinguishing feature of ORION is its support of composite objects. Composite objects are objects with a hierarchy of exclusive component objects. ORION exploits the semantics of composite objects to improve the performance of the system. Its architecture consists of four separate components/subsystems: the message handler, the object subsystem, the transaction subsystem and the storage subsystem. Schema evolution is supported through the screening approach. Concurrency control is based on locking. ORION provides recovery mechanisms for soft crashes and user-initiated aborts. Recovery against hard crashes is not currently supported. ORION provides support for two types of versions: a transient and a working version. ORION supports both forward and reverse query mechanisms. Another prominent feature of ORION is its support for the capture, storage and presentation of many types of multimedia information including sound and video.

2.5.2.2 GEMSTONE

GEMSTONE is an object database product developed and marketed by Servio Logic Corporation of Alameda, California. It provides a back-end object-oriented database environment on a SUN workstation under UNIX or on a DEC VAX under VMS. Typical user access is provided either by an IBM/PC or Macintosh running SMALLTALK/V or a Sun workstation running SMALLTALK/80. Communication is established through TCP/IP, typically with ethernet. GEMSTONE provides forty-four predefined classes and a set of more than six hundred methods [28]. It utilizes the conversion/coercion approach to the management of schema changes. GEMSTONE employs both pessimistic (no conflicting actions) and optimistic (conflicting actions discarded) concurrency control. GEMSTONE does not maintain a log of updates. Therefore, changes that are committed persist while changes not yet committed are lost. GEMSTONE also provides mechanisms for security and authorization. Language support in GEMSTONE is provided by OPAL. The OPAL programming language is based on SMALLTALK/80 and provides data definitions, manipulation and query formulation facilities. It also offers built-in data types, operators and control structures comparable to those provided in PASCAL or C.

2.5.2.3 IDB Object Database

IDB Object database comes from Persistent Data Systems of Pittsburg, Pennsylvania. It is a high performance, distributed object database which allows software developers to define and manipulate objects of arbitrary size and complexity [29]. Objects have attributes (data) and can perform operations (behaviour) and are organized in a multiple inheritance class hierarchy. The IDB Object Database supports classes, multiple inheritance, polymorphism, dynamic binding and data extraction. Concurrency control is implemented by storing locking information in files. Object classes are defined interactively using the IDB Schema Designer and objects' operations are programmed in ANSI C. IDB applications can run on a single machine or across a network.

2.5.2.4 VERSANT ODBMS

VERSANT ODBMS is produced and marketed by VERSANT Object Technology of Menlo Park, California. The VERSANT ODBMS allows the development of applications to manage complex data in distributed, concurrent environments. It runs on a number of platforms which include Sun, DEC, IBM, HP, NCR, Silicon Graphics, Sequent, NeXT and Intergraph as well as IBM PCs and compatibles under OS/2. VERSANT language interfaces include C, C++, Smalltalk and Object SQL. and is also integrated with a wide range of third party

application development products and CASE tools [30].

VERSANT ODBMS is implemented within the VERSANT Scalable Distributed Architecture (VSDA). VSDA means that the ODBMS can be scaled to fit any size organization as needs and situation dictate. There are three keys features of VSDA:

a. *transparent data distribution.* VERSANT is a fully transparent distributed database. This implies that applications can simultaneously connect to multiple VERSANT ODBMS servers on multiple nodes and transparently access the objects managed by those servers. This means that VERSANT can be easily reconfigured and that data can be easily relocated to meet the changing needs of the organization;

b. *advanced workgroup support.* VERSANT includes a number of workgroup functions that exploit the power of today's client-server networks. It includes an object versioning mechanism that allows different groups to concurrently work on different versions of an object and then integrate the different versions; and

c. *schema evolution.* This is the ability to dynamically modify and evolve database schemas so applications can easily scale to the organization's need. VERSANT allows for schema changes "on the fly" without system disruption unlike other ODBMS that have a requirement for a database unload and reload to affect schema changes.

VERSANT includes the following features to maximize performance:

a. *advanced concurrency controls* to guarantee high performance without sacrificing data integrity. Among VERSANT's many concurrency controls are object-level locking, or the elimination of "false waits" and nested transactions;

b. *balanced client-server architecture* that partitions database work across both the client and the server to reduce network traffic, speed performance and fully exploit both desktop and server computing power; and

c. *dynamic disk management* to ensure maximum database performance, especially

P498259.PDF [Page: 49 of 82]

in environment with large objects and complex data. This is done through on-line space reclamation and reuse, and automatic data clustering.

2.6 Computer Security

2.6.1 Technology Description

2.6.1.1 Background

The world today is growing ever more dependent on the ability to retrieve, receive, manipulate and transmit information, be it voice or data, instantly. But as the dependence grows, so too does the need to protect the systems used to execute these operations and their information from unauthorized use, abuse or damage. Nonetheless, the information security industry has not kept pace with the number of information systems. Users are becoming increasingly aware of the security problems and the vulnerability of their data to potential thieves and vandals. Some of the questions being asked are:

a. how to secure telecommunications lines used to enhance the speedy transmission of large transactions?

b. how to manage access to networks and databases by multiple users with varying clearance levels?

c. how to guard against the threat of computer viruses; and

d. how to ensure the integrity of the information on which they depend?

The National Computer Security Center (NCSC) was formed in the US in 1981 to set computer security standards and evaluate systems for use by US government agencies. The NCSC focuses on systems that process classified and sensitive data. The organization develops criteria and guidelines, evaluates hardware and software security properties and evaluates the conduct and support of computer security R&D. To encourage consistency throughout the industry, NCSC introduced guidelines for trusted systems. These guidelines are included in three books:

a. *DoD 5200.28-STD, "Department of Defense Trusted Computer System Evaluation Criteria* [38]. Also known as the "Orange Book", it lists the seven security levels used to categorize an evaluated operating system;

b. *Trusted Network Interpretation*. The "Red Book" is used to interpret the Orange Book guidelines as they pertain to trusted networks; and

c. *Trusted Database Interpretation*. This publication is used to interpret the Orange Book guidelines as they pertain to database management systems.

Once a system is evaluated by NCSC, it is classified into one of seven security levels listed in the Orange Book. These levels are explained in detail in Appendix E.

2.6.1.2 Canadian Trusted Computer Product Evaluation Criteria

In 1988, the Canadian System Security Centre (CSSC) was founded. Its purpose was to generate a series of criteria to address security issues unique to the Canadian government and to establish a Canadian evaluation capability. CSSC has now produced a third version of the Canadian Trusted Computer Product Evaluation Criteria [39]. The Canadian Criteria is divided into two distinct groups:

a. *functionality*. Functionality is defined by four criteria:

1) confidentiality. This defines the ability of a product to protect information against unauthorized disclosure. Confidentiality may be provided through covert channel analysis and the use of discretionary confidentiality, mandatory confidentiality and object reuse services;

2) integrity. This criteria defines the ability of a product to provide data or product integrity. Integrity may be provided by the following services: discretionary integrity, mandatory integrity, physical integrity, roll-back, self-test and separation of duties;

3) availability. Availability defines the ability of the product to control the availability of objects or resources. Availability may be provided through containment, robustness and recovery services; and

4) accountability. This criteria defines the ability of a product to assign responsibility for an action to the proper individual. Accountability may be provided by audit, identification and authentication and trusted path

services; and

b. *assurance*. Assurance is the level of trust put into the system. This criteria includes requirements for architecture, development environment, development evidence, security manuals, security testing and trusted distribution and generation.

Each of the above mentioned criteria is broken down into divisions, each containing levels. A component of a system (functional aspect or mechanism) which offers a level of availability is represented by a functionality division. The Assurance Criteria has only one division which reflects the overall trust the evaluators are putting into the system as a whole. The structure of the criteria is shown in Appendix E.

The government evaluation criteria is the only information security standard available to industry.

2.6.1.3 Computer Security Policy Models

Many computer security policy models exist today. They cover the full range of security policy issues from industrial security where integrity (information is protected from improper modification) is the main concern to military security where confidentiality (information is protected from improper disclosure) is the priority. The most popular computer security models listed below are reviewed in Appendix E.

- Bell-LaPadula (lattice) Model.
- Compartmented Policy Model.
- Multiple Compartmentalized Hierarchy (MCH) Policy Model.
- Government of Canada Security Model.
- Embedded Hierarchies and Compartments (EHC) Policy Model.
- Clark-Wilson Policy Model.
- Biba Integrity Policy Model.

2.6.1.4 Security Certification and Accreditation

P498259.PDF [Page: 53 of 82]

A Designated Approving Authority (DAA) must certify and accredit a system prior to it being allowed to process and handle classified or sensitive information. The DAA should be involved in all the phases of the system acquisition, from the development of the security policy and operations concept for the system to the security testing to ensure the system abides by the security policy and the operational concept.

Certification is the technical evaluation of a system's security features and other safeguards. It establishes the extent to which a system's design and implementation meet a set of specified security requirements.

Accreditation is the DAA's formal declaration, based on the certification process and other management considerations, that a system is approved to operate in one of three security modes:

a. *dedicated*. In this mode, all users have the clearance or authorization and need-to-know for all the data handled by the system;

b. *system high*. In this mode, all users have the security clearance and the authorization, but not necessarily the need-to-know, for all the data handled by the system; and

c. *multilevel*. This mode allows two or more classification levels to be processed simultaneously within the same system when not all users have a clearance or formal access approval for all the data handled by the system.

2.6.2 Trusted Operating Systems

Since the late seventies and early eighties, through its DoD Computer Security Initiative and the DoD Computer Security Evaluation Center, the US Department of Defense (DoD) has attempted to raise general awareness of computer security issues and to urge manufacturers to improve the security of commercial operating systems offerings. These efforts have been rewarded with the appearance of various secure computer operating systems on the market. Description of some of the most notable ones are provided in Annex D.

P498259.PDF [Page: 54 of 82]

3. APPLICATION TO LAND ELECTRONIC WARFARE

3.1 Problem Areas

The objective of Land Electronic Warfare (Land EW) is to exploit the radio frequency spectrum to assess the location and intentions of opposing forces. Through this exploitation, friendly forces may take effective action.

The major activities associated with the achievement of this objective encompass:

1) Detection and Location - The detection and direction finding of opposing force emitters.

2) Intercept & Identification - Through interception of comms and parametric analysis of emitter characteristics the identification of the user of the opposing force emitter.

3) Analysis - Through geometric and correlative analysis, determine the electronic order of battle (EORBAT) and intentions of the opposing force.

4) Reporting - Presentation of the analysis with supporting data in a timely manner such that supported formations may take effective action to counter the intent of the opposing force.

Current fielded equipment and personnel can effectively perform these activities only for a low to moderately intense conflict. In a high intensity conflict with a dense emitter environment, the Land Electronic Warfare system could be quickly overwhelmed by the mass of data and, worse, unable to differentiate or prioritize threats.

It is anticipated that, in an effort to avoid detection, location and intercept, opposing forces will resort to increasingly more exotic communication technologies. By the year 2000, it is likely that only 10% of emissions will be clear-voice. The remainder will be digital transmissions of data or voice and that much of this will also be encrypted. Significant use will be made of spread spectrum, frequency hopping and other hard-to-detect emitters such as cell phones.

Low bandwidth availability on the friendly nets will likely continue to be a problem. In a high threat density environment, it will likely not be possible to simply transmit raw data from the sensors to the EW analysis position. Instead, only the information germane to the performance of the analyst function may be sent.

Several elements of the EW process must be improved to overcome these difficulties. Specifically:

Sensor systems must be able to differentiate incoming signals automatically by frequency and direction even in the presence of multiple signal events occurring at the same time;

Identification of emitter types on the basis of parametric data should be performed automatically if the emitter type is known;

Having due regard for terrain, emitter types and known characteristic (doctrinal) EORBAT, the system should present hypotheses regarding the likely matches between detected emitters and available doctrinal templates with a confidence factor;

If an hypothesis is accepted by the analyst, the system should present options for confirming the hypothesis by suggesting new sensor taskings and recommending new sitings for sensors having due regard for battlefield and terrain constraints;

Both hypotheses presentation and confirmation options should be presented with the option for the analyst to review the logic which led to the presentation;

When an analyst is ready to present his findings, the system should assist in gathering the data supporting the assessment and should support the communication of that assessment and data in the most economical manner to reduce bandwidth usage and to limit opportunities for opposing forces EW assets to detect the transmission;

The system should assist intercept operators in transcribing intercepts (spoken word recognition) and in identifying speakers; and

Processing of terrain data should be extended to include surface feature data such as vegetation and buildings.

P498259.PDF [Page: 56 of 82]

A corollary requirement for the enhanced Land EW capability is the development of a simulator which can present realistic scenarios and which can react appropriately to the actions of the participants.

3.2 Technology Solutions

The technologies discussed in section 2, can be applied to meeting the challenges described in section 3.1. Acousto-optic processors show promise of being able to perform broad-band spectral processing (search) and direction finding in a more direct and less computationally intensive manner than conventional systems. This is particularly important in a dense emitter environment.

The output of an Acousto-Optic Processor is an electronic array of values which can be input to a computer as a bit stream created by scanning an array of charge-couple devices. This method is slow enough that it may defeat the speed gains made by using the AO processor.

Neural Networks implemented on a parallel processor arrangement could read and process the data arrays at the same time. Properly trained, a neural network could accurately differentiate multiple signals while discarding side lobes and harmonics. The output of the neural net would be the captured frequency and either a line of bearing to the emitter or a DF on the emitter (see example three in section 2.2). Parallel computers could be applied to propagation path reconstruction calculations to reduce the Circular Error Probability of the fixes through the application of surface feature data.

In a dense emitter environment, the use of AO processors, neural networks and parallel processors, as so far described, increases the probability of detection, the accuracy of the detection and the parametric information available. However, these solutions have done nothing to assist the intercept operator in transcription of intercepts nor the analyst in determining the EORBAT of the opposing force. In a dense environment the shear volume of this accurate information could, still, so overwhelm the analysts and operators as to make it impossible to assess the situation quickly enough for the assessment to be of value.

Parallel Processing, along with Expert Systems and databases can be of assistance at this point in the EW process. As noted earlier, at least two research projects are investigating direct voice to text applications. They require the kind of computing power available only in super computers and parallel processor configurations. It should be noted that, for EW

P498259.PDF [Page: 57 of 82]

purposes, the domain of words which must be recognized at any time is much smaller than that for which these systems are being developed and therefore, less computing power may be required.

As noted in section 2, the power of parallel processing has also been applied to voice recognition systems. If this support were provided to intercept operators it could significantly enhance the value of their efforts.

Automatic correlation of DF's with intercept on the basis of time, frequency and electronic visibility (Propagation Path re-construction) adds yet another dimension to support data analysis. Dependent on the available power, this function could be performed on either the DF or intercept parallel processor or may require a separate processor.

To further reduce the amount of information which the analyst must deal with, an Expert System must be used to attempt to match the situation as expressed in the DF and Intercept information, with known EORBAT templates. The templates, must be stored in a database which correlates doctrinal and technical information so that the Expert System can attempt matches on the basis of both geometric (location and terrain) and emitter parametric information.

By presenting the hypothesis with both confidence factors and explanations the system has reduced the problem for the analyst to a manageable level. The use of templates also allows the system to request prioritization for targeting missing elements. Expert Systems can readily be developed which can then present candidate options from which the analyst can choose appropriate tasks to affect the re-targeting.

When the process has cycled for long enough to give the analyst a reasonable confidence in the assessment, both the accepted hypothesis and the supporting data can be packaged by an Expert System in such a way as to reduce the data to the essentials necessary for recreation at the receiving end by a counterpart ES.

Figure Figure 11 provides a block diagram noting the application of the various technologies to the components of the EW process.

In the remainder of this paper technology applications for sensors will only be briefly addressed. The main concentration of the paper will be on the application of the

P498259.PDF [Page: 58 of 82]

technologies to the analysis and comand and control functions.

P498259.PDF [Page: 59 of 82]

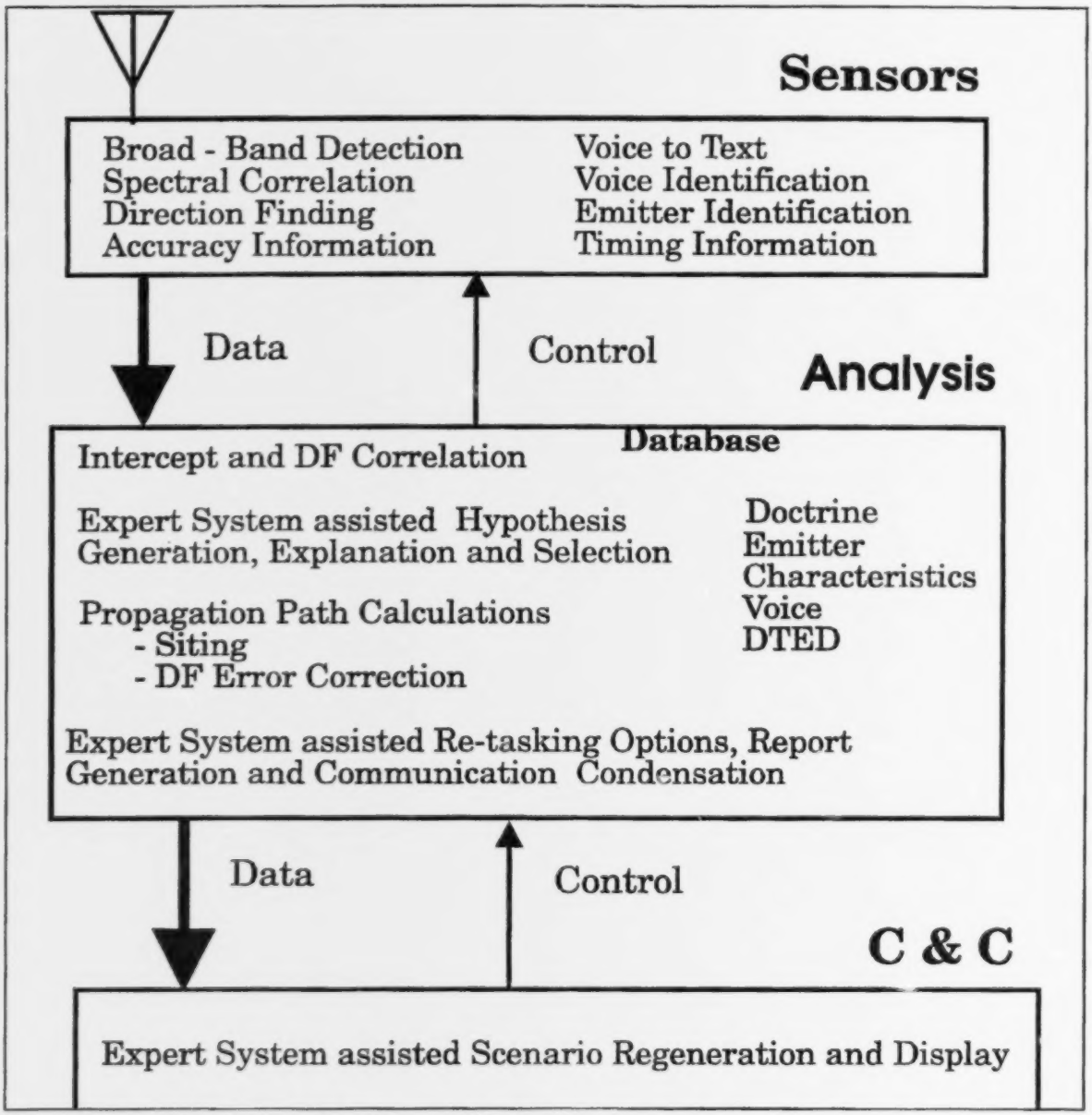

Figure 11 Technologies of an Advanced EW Process

4. RESEARCH AREAS

Section 2 described the state of advanced technologies applicable to the EW process and section 3 described how these technologies might be applied to improve the process in a dense emitter environment. This section will identify and describe the weaknesses of the technologies which must be researched to make practical the application of the technology to the EW problem.

4.1 Sensor Technologies

Several technologies could be applied to improve sensor capabilities. Under laboratory conditions, Acousto-Optic processors have shown promise of being able to meet many of the challenges of successfully performing emitter location and broad band spectral processing. Significant research will have to be performed to achieve a field-able system. This research will be required in areas of compact very broad band Bragg Cells, low energy-use lasers with matching detector arrays and rigid light-wieght light devices (mirrors, lenses).

Neural Networks could be applied to three areas of sensor technology. Neural Networks could be used for emitter identification in a dense environment as well as for voice-to-text transcription and voice recognition. Research would have to be performed in the development of highly adaptable neural networks so that they can be trained quickly according to the situation expected on the battlefield. Further research would have to be performed on the development of databases that would serve as the training data. These databases would have to contain training data on emitter characteristics, languages (for voice-to-text) and voice characteristics of specific voice targets. Developing, maintaining and extending these databases would be a substantial effort.

4.2 Analysis and Command and Control

4.2.1 Information Processing Technologies

4.2.1.1 Parallel Processing

The key issues in exploiting the power of parallel processors:

a) determining the type of parallel processor architecture for the purpose (SIMD vs MIMD); and

b) for MIMD parallel processors determining the memory architecture (shared vs distributed) dependant on the need for scalability, the granularity required, availability of software environments and cost/benefits..

Most requirements for parallel processors (for analysis and command and control) require a Multiple Instruction Multiple Data (MIMD) architecture. Research will have to be performed to determine whether these should employ shared or distributed memory. The pros and cons of these choices are described in section 2.1. If it is determined that shared memory is required, then research will be required into methods for limiting bus congestion. If distributed memory parallel processors are required, research will have to be done in two areas:

1) Programming Languages and Environments for the development of parallel software; and

2) Communication and Switching Technologies.

4.2.1.2 Expert Systems

There are numerous areas of application of Expert Systems which must be researched. Specifically:

For Analysis and Command and Control:

Hypothesis Generation;
Templating;
Retasking Options Generation;
Propagation Path re-construction (CEP reduction);
Assessment Condensation & Regeneration; and

For a Simulator:

On-the-fly scenario modification.

Part of this research will involve determining which architecture (Semantic Network, OVA Triplet, Rules, Frames or Logic) is most appropriate to the requirement for each application area.

All expert applications require a base of rules or procedures which define the logic to be employed in arriving at a solution to the problem. While systems exist that can assist a knowledge engineer in determining the rules to be employed, significant research is still required to determine whether a rule set is complete and consistent. That is: that a valid solution will always be found when one exists and that all solutions found are, in fact, valid.

Currently, most Expert Systems are extremely memory and computationally intense. The memory intensity can be lessened by having the Expert System interact with a database. However, for this to be efficient, there must be a very tight coupling between the Expert System engine and the database engine. Problems resulting from the computational intensity of Expert Systems have, so far, only been resolved by employing greater computer power. Because Expert Systems problems can generally be resolved in parallel, the computational power requirements problem can be resolved using parallel processor systems. There is, however, still significant research required into discovering the most efficient mechanisms for distributing the problems to the individual processors so that loads are balanced and the resolvants are returned in a coherent manner.

4.2.1.3 Object Database Management Systems (ODBMS)

The EW function processes information about objects and collections of objects (i.e. a unit is a collection of one or more emitters which is defined from a collection (a cluster) of DF hits). This kind of assembly is called, in database terms, a parts explosion. Traditional (relational, hierarchical) databases and linear languages (C, Ada, etc.) experience significant difficulty in dealing efficiently with this concept. This was one of the reasons that an Object-Oriented (O-O) approach was taken in the development DFACTT and the fielded CEWOC system. Other large binary EW objects such as voice, graphics, and video are not handled well in a non-ODBMS.

Object Database Management Systems (ODBMS) are just beginning to be marketed and the available products are very uneven in their useability. Even if a product meets all of the mandatory requirements described in annex D, the buyer must still determine which of the optional choices are desirable/available and the validity of the choices made by the implementor in the open choices area. A fairly straight forward market survey can answer most of these questions.

However, this knowledge must be coupled with knowledge about the intended environment. In the EW case, there are two, potentially conflicting, requirements which the ODBMS must meet.

1) Most of the data currently available regarding technical characteristics of battlefield emitters is stored in RDBMS's. The ODBMS which must go to the field must be able to accept this data and create useable objects from it. Of equal importance, if a new emitter type is discovered during operations, the technical data must be exported to the RDBMS.

2) Given any model of the technology in the EW environment similar to Figure Figure 11, the ODBMS must be able to operate in a parallel computer platform and must efficiently respond to demands from at least Expert Systems application and, O-O applications dealing with potentially large collections of possibly large (Voice, Graphics) base object types.

Research needs to be conducted into efficient ways in which to interface ODBMS's to relational databases. Some ODBMS have an RDBMS engine at their core but this does not, necessarily, imply that a relational interface is available to support data transfer, nor does it mean that a simple transfer of relational data would result in the creation of useable objects.

While some ODBMS support the concept of distributed objects across various platforms, it is unclear whether any could be efficiently employed on a tightly coupled parallel architecture. Since the demands made by a Expert System on its database can (and in the EW environment will) be very intense, this requirement must be met. Given the fairly immature state of ODMBS technology it is certain that extensive research will be required in this area.

The nature of objects which are to be stored in ODBMS must also be given significant consideration in the design of this system. Among the questions which must be considered are:

> Must this object be persistent (i.e. does it need to live beyond the current execution of the application) or must it only live for some definable period (possibly the whole life of the current execution of the application)?
>
> How often must the object be accessed? Some ODBMS are more efficient of access than others but have other, perhaps less desirable, attributes.
>
> Are the objects which must be persistent mostly data, mostly behaviour or some balance of the two? Different ODBMS handle these characteristics differently and are more efficient with one type than the other.

4.3 Computer Security

Certain components of the data which will be resident on EW systems and, indeed, some of the algorithms may be highly sensitive and must be protected from unauthorized disclosure and damage. In determining the security requirement one must consider the threats to the assets, the risk that the threat will occur, and what measures may or must be taken to reduce the risk.

The threat to disclosure include an authorized user of the system who is not authorized to see particular data and, in the case where a unit has been overrun, an unauthorized person capturing an intact system.

Threats of damage and destruction are far more numerous. If the system does not have adequate safe-guards an authorized user can inadvertently damage data or, in fact, the application, making it unusable. If detected, any number of weapons can be used to destroy or incapacitate the system or users.

A more detailed threat, vulnerability assessment (TVA) could, no doubt, detect numerous other serious threats and risks.

To mitigate the risks mentioned above, several actions can be taken. To reduce the risk from

inadvertent damage, a development regime which requires a significant effort in insuring the quality of the software must be developed. While such regimes exist for some linear languages, the research in metrics and standards in the O-O realm is still in its infancy.

Overt acts leading to unauthorized disclosure can be countered by employing secure operating environments and through user actuated destruction mechanisms. If the software solution (secure operating environment) is to be employed, a determination of which systems will require protection must be made. Most effort in the development of secure operating systems has concentrated on mono-lithic (single processor) systems or loosely coupled multiprocessor systems. Research on operating systems for parallel processors has concentrated on making efficient use of the available power. A significant research effort would have to be made to achieve a secure operating system for a parallel processor.

Perhaps the best protection against overt action is detection and intercept avoidance. The greatest threat to detection avoidance is from opposing force EW assets. To this end, several areas need to be explored. Hiding the transmissions through low power spread spectrum or frequency hopping is a good measure. However, one must assume that the opposing force will have equipment at least as good as our own and the research recommended so far could defeat these measures if any transmission was of sufficient duration. However, if each transmission is sufficiently short, and employs the above measures (spread spectrum etc.) then it will be extremely hard to detect.

Since an Expert System is being employed to assist in developing the situation assessment, it can determine the minimum set of data which would be acquired to recreate the assessment. It can also package the information for transmission into a set of compact messages which would require the friendly emitter to be on air for the shortest possible time. To be effective a complimentary Expert System would have to be at the receiving end to re-create the assessment. This will require research into efficient forms of communication, as well as determining how an Expert System can decide on the minimum set of data.

5. Research Requirements

This section provides a re-statement of each research area identified in the previous section together with a statement of the risks, benefits and estimated costs of such research. Risks and benefits are provided as ratios in accordance with figure Figure 12. For example a ratio of 5:3 would indicate a high risk, moderate benefit, while a ratio of 2:4.5 would be relatively low risk with quite high benefits.

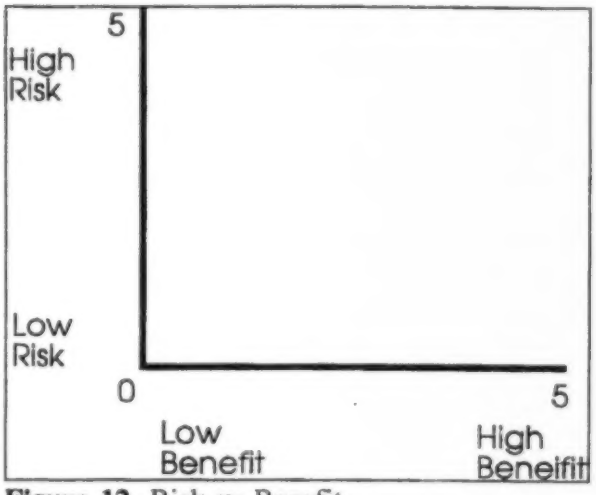

Figure 12 Risk vs Benefit

5.1 Sensor Technology

5.1.1 Acousto-Optic Processors

5.1.1.1 Research

The required research activities in AO processors are:

1) Improvements in Bragg cells to obtain very low acoustic velocity materials;

2) Very high light intensity, low input power, low heat lasers and matching high speed detector arrays; and

3) Stable light guides (mirrors) and focusing elements for a possibly polluted environment.

5.1.1.2 Risks

All three of the research areas must achieve significant success to make the use of AO processors feasible in the EW environment. Failure in any area would result in a processor which was too bulky, too power consumptive, too slow or too unstable.

5.1.1.3 Benefits

The benefits of success in AO processor research are significant. A relatively small number of sensors using this technology could provide broad-band detection, DF and parametric information against a wide assortment of exotic emitters.

5.1.1.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in these research areas, effort should be applied as follows:

1) Exploitation of Bragg Cell research - 5 - 10 PYs - Ratio: 5:4;

2) Exploitation of Lasers and Detector Arrays - 5 - 10 PYs - Ratio: 3:4; and

3) Light Guides and Focussing Elements - 5 - 10 PYs - Ratio: 3:4.

Overall Risk - Benefit Ratio: 3.5:4

5.1.2 Neural Networks

5.1.2.1 Research

The required research activities in this area are:

1) Training of Neural Networks as Array processors and for use in voice-to-text and speaker identification functions; and

2) Development of training databases.

5.1.2.2 Risks

Considerable work has been done in the area of speaker-independent speech recognition for control of computers. Converting this for use in a voice-to-text application should be achievable though doing so for many languages presents a significant challenge.

Training a Neural Network to discriminate amongst and parameterize multiple simultaneous emitters is likely to be extremely challenging. This is also true with respect to using Neural Networks to perform voice identification.

Development of training databases to support Neural Networks must be done in concert with the development of the Neural Networks since the data required will likely not be completely known until the Networks have been developed.

5.1.2.3 Benefits

The potential benefits from the use of these technologies are significant. In the area of Neural Nets, if only voice-to-text were achieved, the burden on Intercept operators would be greatly reduced. This would allow them to more effectively capture greater numbers of communications and, thus be more likely to identify key targets.

If Neural Networks can also be used to discriminate amongst and parameterize multiple emitters, the Analysts' burden will also be reduced since effort can be shifted from emitter identification to unit identification and situation assessment.

P498259.PDF [Page: 69 of 82]

5.1.2.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in these research areas, effort should be applied as follows:

1) Exploitation of Neural Networks (Speech to text and Voice Identification) - 5 - 10 PYs - Ratio: 4:3;

2) Exploitation of Neural Networks (Emitter discrimination and parameterization) - 10 - 15 PYs - Ratio: 4:3; and

3) Training database development - 15-20 PYs - Ratio: 4:3.

Overall Risk - Benefit Ratio: 4:3.

P498259.PDF [Page: 70 of 82]

5.2 Analysis and Command and Control

5.2.1 Parallel Processing

5.2.1.1 Research

The required research activity in this area is:

1) MIMD architectures, languages and communication technologies.

5.2.1.2 Risks

The promise of significant cost improvements of parallel computers means that industrial research is delivering continuous improvements in the development of more powerful systems. The development of languages and tools which would allow programmers to make efficient use of this power has not, however, kept pace. The tendency is to tailor languages to the specific machine, host operating system and application. This is expensive and a significant effort will be required to develop more general purpose languages that can operate across a variety of host architectures and operating systems.

5.2.1.3 Benefits

The development of efficient parallel processor languages will mean that better algorithms to support the entire EW process will be more easily achieved.

5.2.1.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in this research area, effort should be applied as follows:

1) Exploitation of Parallel Computer Languages - 15 - 20 PYs - Ratio: 2:5.

5.2.2 Expert systems

5.2.2.1 Research

The required research activities in this area are:

1) Mechanisms for assuring that a knowledge base is complete and consistent;

2) Development of expert systems and their integration onto the selected platforms; and

3) Application Techniques. The application areas identified at the beginning of section 4.3 will all require algorithms for pattern matching and planning. Such algorithms must be efficiently implemented or they will take too long to execute to be effective.

5.2.2.2 Risks

Determination that a rule-base (knowledge-base) is complete and consistent is a mathematically hard problem. Certain classes of rule bases defy mathematical treatment and developers (knowledge engineers) must rely on heuristics and reviews to provide assurance of completeness. Current research in this area is quite extensive and some, relatively simple, problems have been solved. However, some problems have been proven to be insoluble in finite time and this presents a risk that there may be no way to be assured, mathematically, that an Expert System rule base for the EW process is complete. This risk is mitigated by the fact that the EW support system does not have to be perfect, only very good.

Expert System shell are quite complex pieces of code (The C-Prolog interpreter is nearly one million lines of C source code). Re-working such a system to make efficient use of a parallel processor is not risk-free. Before starting such a venture, there must be assurance that the supported shell meets the application criteria.

The application requirements are all reasonably well-understood. All are also quite difficult to implement since all must be performed under near-real-time conditions.

5.2.2.3 Benefits

If the rule-base can be made, at least heuristically, complete and consistent, then the analyst will be able to use his own knowledge and experience to pick an hypothesis and re-tasking options from among those presented. This will reduce the effort of the analyst in developing the assessment and allow him to pay more attention to assuring himself that a correct assessment of the situation has been made. With this support, he can concentrate on examining data in areas where he is unsure of the assessment instead of concentrating on assembling the data.

Propagation path reconstruction (CEP reduction), if accurate enough, could permit targeting of emitters with out undue expenditures of munitions.

Expert Systems which can reduce the amount of data required to transmit the assessment and to perform regeneration of the assessment at the receiving end, could significantly reduce the on-air time. This, in turn, reduces the risk to ones own assets. It is possible that developments in this area for the EW environment could have wider implications in other areas where band-width is scarce or where communications security is at risk.

A simulator is needed to support economically viable training of EW operators for work in a dense emitter environment. The simulator must not only presents a scenario but be able to modify the scenario on-the-fly to reflect the actions of the EW operators and other participants in the training. Research in Expert Systems for the EW process would benefit the development of such a simulator.

5.2.2.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in these research areas, effort should be applied as follows:

1) Knowledge base completeness/consistency - 5 - 10 PYs - Risk: 2:4;

2) Integrations of Expert Systems - 10-15 PYs - Risk: 2:4;

3) Application Techniques - 15 - 20 PYs - Risk: 1.5:4.5;

Overall Risk - Benefit Ratio: 2:4.

5.2.3 Object Database Management Systems

5.2.3.1 Research

The required research activities in this area are:

1) Effective interfaces between the ODBMS and other conventional databases;
2) Effective/Efficient interfaces to Expert Systems and OO languages; and
3) Operation of the ODBMS on the selected processing architecture.

5.2.3.2 Risks

Development of a straight-forward interface to a relational database (that is: importing and exporting of data) is a relatively low-risk under-taking. However, for each entity of interest in the relational database, a class of objects with appropriate behavioral characteristics must also be developed. This is more complex. To have the skeleton of the object class created automatically would require a substantial research effort.

Because ODBMS are designed to manage objects, interfaces to OO languages already exist and extensions to handle a different OO language should not be extraordinarily difficult. However, an interface to an Expert Systems language could be significantly more difficult. The programming paradigm for most Expert Systems is quite distinct from the OO paradigm. As a result, there could be considerable difficulty in development of such an interface.

Many ODBMS are designed to operate in a distributed (loosely coupled) processor environment. It is unknown how efficient (or even useable) the communications scheme from such a system would be in other environments. It is conceivable that a completely different communications architecture would be required for a parallel processor environment.

5.2.3.3 Benefits

The ability of the EW system to effectively upload data from conventional databases is essential to providing EW operators with basic information about opposing forces' doctrine, tactics, the technical characteristics of their equipment, their language and the voice prints of key personnel. Availability of this information for use by the systems described earlier and directly by the operators can significantly improve the efficiency and quality of the EW process.

As noted earlier, interaction with a database is generally essential to the effective operation of an expert system. Thus efficient interfaces will provide benefits in almost all the application areas described in section 4.3.

It is imperative that the ODBMS not create a bottle-neck for the operation of other systems. To avoid this, the ODBMS must respond quickly to the demands placed on it. Thus, it must operate in a parallel computing environment.

5.2.3.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in these research areas, effort should be applied as follows:

1) Interfaces to conventional DBMS - 5 - 7 PYs - Risk: 2:4;

2) Interfaces to OO and Expert Systems - 10 - 15 PYs - Risk: 2:4 ; and

3) Operation on the chosen architecture - 10 - 15 PYs - Risk: 3:4.

Overall Risk - Benefit Ratio: 2:4.

5.3 Security

5.3.1 Research

The required research activities in this area are:

1) Development of a rugged development environment to limit the risk that a user will, through inattention or malicious intent, be able to damage the data or applications;

2) Development of Trusted Operating Systems and applications for parallel processors; and

3) Communication Data Reduction (Complementary to the Expert System research).

5.3.2 Risks

A development environment cannot prevent a programmer from writing bad code. A good environment can, however, assist programmers and program managers in assessing the quality of the code. Unfortunately, development of quality metrics for OO code is still fairly limited. Furthermore, there is still significant disagreement in the OO community with respect to defining what the quality factors are. Thus, the risks to the successful development of a development environment supporting high quality programming are substantial.

There have been several successful implementations of trusted operating systems and applications at the 'B1' level of trust. These have all been for operation on monolithic computers and all have been very expensive to develop. The likelihood that any company would see a market for this kind of development for parallel processors is quite remote. The risks in trying to develop such an environment without an observable market are very high.

While the research in the expert systems domain will lead to reducing the number of data elements which needs to be transmitted, other technologies may be employed to efficiently encode the data to further reduce the on-air time. The search for such algorithms has been on-going for many years in response to increasing demand for information transfer through narrow band-width pipes. Generally speaking the trade-offs are between reducing the number

P498259.PDF [Page: 77 of 82]

of bits transferred and the processing power needed to encode and decode the data.

5.3.3 Benefits

The benefit of developing a rugged development environment for the delivery of high quality code extends beyond the realm of the applications needed for the EW process. Assurance that programs will do exactly what they are designed to do and which will properly protect themselves and their data is a goal of most development efforts.

Systems operating trusted applications under the control of a trusted operating system on a parallel processor would substantially reduce the vulnerability of the system to either inadvertent or malicious damage.

From a security stand-point, reduction of on-air time is always a benefit. Thus research in this area could reap substantial benefits for both EW systems and for other forces seeking to avoid EW detection.

5.3.4 Costs and Risk - Benefit Ratios

It is estimated that, for a reasonable chance of success in these research areas, effort should be applied as follows:

1) Development of a development environment - 20 - 30 PYs - Risk: 3:4;

2) Trusted operating systems and applications - 150 - 200 PYs - Risk: 4:1; and

3) Communication Data Reduction - 10 - 15 PYs - Risk: 2:4.

Overall Risk - Benefit Ratio: 3:3.

6. Recommendations

The following table provides a recap of the research requirements, the level of effort required and the associated risks. Risks and benefits are provided as ratios in accordance with figure Figure 12. For example a ratio of 5:3 would indicate a high risk, moderate benefit, while a ratio of 2:4.5 would be relatively low risk with quite high benefits.

Research Description	Level of effort leading to success	Benefit / Risk
SENSOR TECHNOLOGIES		
Acousto-Optic Processors	15-30 PYs	3.5:4
Exploitation of Bragg Cell research	5 - 10 PYs	5:4
Exploitation of Lasers and Detector Arrays	5 - 10 PYs	3:4
Light Guides and Focussing Elements	5 - 10 PYs	3:4
Neural Networks, Array Processors and Parallel Computers	30-45 PYs	4:3
Neural Networks (Speech to text and Voice Identification)	5 - 10 PYs	4:3
Neural Networks (Emitter discrimination and parameterization)	10 - 15 PYs	4:3
Training Databases	15-20 PYs	4:3
INFORMATION PROCESSING TECHNOLOGIES		

Research Description	Level of effort leading to success	Benefit / Risk
Parallel Computer Languages	15 - 20 PYs	3:4
Expert Systems	30 - 50 PYs	2:4
Knowledge base completeness/consistency	5 - 10 PYs	2:4
Expert Systems integration	10-20 PYs	2:4
Application Techniques	15 - 20 PYs	1.5:4.5
Object Database Management Systems	25 -37 PYs	2:4
Interfaces to conventional DBMS	5 - 7 PYs	2:4
Interfaces to OO and Expert Systems	10 - 15 PYs	2:4
Operation on the architecture	10 - 15 PYs	3:4
Security	180 - 245 PYs	3:3
Development of a development environment	20 - 30 PYs	3:4
Trusted operating systems and applications	150 - 200 PYs	4:1
Communication Data Reduction	10 - 15 PYs	2:4

It is recommended that research effort be prioritized on the basis of the benefit/risk ratios. It should be noted that use of parallel processors and ODBMS provide the greatest overall benefit to the EW process.

P498259.PDF [Page: 80 of 82]

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Software Kinetics Ltd.
65 Iber Road
Stittsville, Ontario K2S 1E7

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

~~Unclassified~~
UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

DFACTT Data Fusion Tools and Techniques Report on Advanced Technologies (U)

4. AUTHORS (Last name, first name, middle initial)

Dumoucel, Luc
Crabb, Cheryl
George, Art

5. DATE OF PUBLICATION (month and year of publication of document)

March 1995

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

129

6b. NO. OF REFS (total cited in document)

44

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defence Research Establishment, Ottawa
Shirleys Bay, Ontario

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

W7714-2-9656101-QC

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

1600-116-006 Version 01

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

DREO CONTRACTOR REPORT 95-644

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

- ~~( ) Unlimited distribution~~
- (X) Distribution limited to defence departments and defence contractors; further distribution only as approved
- ( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
- ( ) Distribution limited to government departments and agencies; further distribution only as approved
- ( ) Distribution limited to defence departments; further distribution only as approved
- ( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited distribution

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Technologies which may have application in the area of Electronic Warfare are discussed and assessed in terms of research risk and benefits. The technologies discussed include:

Parallel Processing;
Acousto-Optic Processing;
Neural Networks;
Expert Systems;
Object Databases; and
Computer Security.

Methods for balancing strength and weaknesses of the technologies for the EW application are discussed.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selecte from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Acousto-Optic Processor
Lasers
Bragg Cells
Detection Arrays
Acousto-Optic Signal Acquistion
Spectrum Analysis
Direction Finding
Multiple Instruction Multiple Data
Single Instruction Multiple Data
Hypercube
Systolic Array
Object-Oriented
Crossbar Switch
Omega Network
Summation Function
Transfer Function
Weights
Supervised Learning
Back Propagation
Recurrent Back Propagation
Problem Solving
Prediction
Diagnosis
Planning
Design
Semantic Network
Object-Value-Attribute
Rules
Frames
Logic
ODBMS Manifesto
Data Fusion and Correlation
Orange Book
CTCPEC
Security Policies
Compartments
Trusted Operating Systems